FCAT: A FLEXIBLE CLASSIFICATION TOOLBOX FOR SIGNAL DETECTION IN HIGH-THROUGHPUT SEQUENCING DATA

by

Bing He

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

April, 2017

Abstract

As applications of high-throughput sequencing technologies continue to grow at a fast rate, being able to conveniently develop effective data analysis solutions that can take full advantage of application-specific data characteristics is becoming increasingly important. FCAT is a flexible classification framework and toolbox for signal detection in a wide class of high-throughput sequencing applications where the objective is to locate signals in the genome based on their enrichment, shape and other features. FCAT takes aligned sequence reads (BAM files) as input. It uses supervised learning to automatically extract application-specific features that distinguish signals from noises. Users can aggregate multiple learning algorithms including random forests, L1- and L2-regularized logistic regression to improve prediction accuracy and robustness. A non-parametric inference method is developed for estimating false discovery rate of prediction results. We demonstrate FCAT through a variety of applications including analyses of DNase-seq, ATAC-seq, ChIP-seq, GRO-seq and TIP-seq data. We show that FCAT not only offers flexibility and convenience to handle data from different sequencing applications, but also yields competitive or

ABSTRACT

improved signal detection accuracy compared to existing tools for each application. The FCAT framework can greatly increase the efficiency and reduce the burden for developing bioinformatics solutions to new sequencing applications. FCAT is an open source software package developed using C++ and Python. It is freely available at `https://github.com/HeBing/FCAT`.

Adviser: Dr. Hongkai Ji

Thesis Committee:

Dr. Jiang Qian (Chair)

Dr. Hongkai Ji

Dr. Ingo Ruczinski

Dr. Jiou Wang

Acknowledgments

The completion of this dissertation is the milestone of my PhD research and I owe my gratitude to all who have helped me along the way.

My deepest gratitude goes to my advisor, Dr. Hongkai Ji. I have been very fortunate to have an advisor who is so supportive of and patient with students. He has been always there to talk to, provide suggestions, and give encouragement. I learned so much from him about the philosophy of research, the art of scientific writing and presentation, and the ways of communicating with collaborators. I am deeply grateful to him for supporting me through the PhD program.

I am also very thankful to Dr. Ingo Ruczinski. I would like to thank him for being a member of both my proposal defense committee and my thesis defense committee and for giving me valuable comments on my proposal and thesis.

I would like to thank Dr. Jiang Qian, Dr. Jiou Wang, Dr. Zhibing Wang and Dr. Abhirup Datta for serving as members on my thesis committee. I benefited much from their comments and feedback on my thesis.

I am also grateful to Dr. Kathleen H. Burns, who introduced me to the exciting

analysis on transposon and granted me the access to an amazing dataset TIP-seq. She helped me understand the details of the experimental design and data collection process and has provided valuable insights for the conduct of this research.

I would also like to thank Dr. Ciprian M. Crainiceanu and Dr. Vadim Zipunnikov, who provided much support and guidance in the early years of my PhD study.

I also want to express my gratitude to Weiqiang Zhou, Zhicheng Ji and Dan Jiang for their encouragement and valuable practical advice during my PhD study. Many other friends also have helped me make it through setbacks and stay focused on my graduate study. I deeply appreciate their support.

Finally, I want to thank my parents, my husband Mao and my son Harvey, who have always been standing by my side and believed in me. The love, care and support provided by them is the source of my courage. I want to convey my heartfelt gratitude to them.

Dedication

This thesis is dedicated to my family.

Contents

List of Tables

List of Figures

Chapter 1

Overview

This dissertation presents a flexible classification toolbox for signal detection in high-throughput sequencing data (FCAT). It is organized as follows: Chapter 1 will provide the overview and related literature. Chapter 2 presents the pipeline and methods of FCAT. Chapter 3 discusses applications of FCAT in prediction transcription factor binding sites; applications of FCAT in GRO-seq and TIP-seq are discussed in Chapter 4 and 5, respectively. Chapter 6 summarizes FCAT, discusses limitations of FCAT and provides future directions. Chapter 7 includes supplementary materials.

1.1 Introduction

High-throughput sequencing technologies are widely used in biomedical studies to collect genomic and functional genomic information required for understanding

complex biological systems. In the past decade, a variety of applications of these technologies have been developed. They allow investigators to examine a biological system from multiple complementary perspectives. A few examples include RNA sequencing (RNA-seq) for measuring transcriptome,[1] chromatin immunoprecipitation followed by sequencing (ChIP-seq) for mapping transcription factor binding sites and histone modifications[2,3] , DNase I hypersensitive site sequencing (DNase-seq)[4] and sequencing assay of transposase-accessible chromatin (ATAC-seq)[5] for locating active cis-regulatory elements, bisulfite sequencing (BS-seq) for analysing DNA methylation,[6] and global nuclear run-on sequencing (GRO-seq) for assaying the genomic location and rate of RNA production in nuclei.[7]

Today, new applications of sequencing continue to emerge at a fast rate. Increasingly, how to effectively meet the growing demands of data analysis becomes a challenge. Each sequencing application will generate data with its own unique characteristics. A powerful data analysis solution should take full advantage of these application-specific characteristics. Traditionally, data analysis solutions are developed separately for different applications. For each application, one or multiple bioinformatics experts are recruited, and they will spend tremendous amounts of time on developing models and algorithms tailored for that specific application, implementing them in computer programs, and debugging and testing these programs to make sure that they are optimized and robust. Many methods and software tools are developed using this approach. Examples include tools developed for peak calling in ChIP-

seq data,[8–20] predicting transcription factor binding sites (TFBS) from DNase-seq data,[21–24] and identifying active transcriptional regulatory elements from GRO-seq data,[25] etc.

Increasingly, however, this case-by-case approach becomes inefficient to meet the diverse data analysis needs. There are multiple reasons for this. First, developing specialized tools for each and every sequencing application requires investment of significant amounts of developers time and resource which are not always available. For example, many small laboratories that produce data do not have enough resource to support a full time method developer devoted to their projects, nor does every laboratory have access to an experienced method developer with the required knowledge and interest to handle the data. For method developers, they may have many other responsibilities that constrain their ability to devote large amounts of time to a specific application. Together, these factors can cause significant delays in developing optimal data analysis solutions for a new data type or application. Second, some specialized sequencing technologies and applications are only used by a small number of laboratories. New experimental techniques under development may also only have a very limited user base before they mature. While data generated by these technologies and applications can be highly valuable, the limited user base may not create a strong incentive for method developers to invest tremendous amounts of time to develop and optimize data analysis solutions. This creates a dilemma because without an optimized data analysis tool the data may not be used effectively. Third,

although different sequencing applications have different data characteristics, many of them share a similar data structure and analysis goal. For example, both ChIP-seq and DNase-seq data analyses involve detecting genomic locations with enrichment signals. Developing a separate solution for each application will inevitably involve reinventing the wheel to certain extent. Developers may need to write similar codes repeatedly for data pre-processing, feature extraction, and other similar procedures. It does not represent the most efficient use of time and resource.

To help better meet the growing data analysis needs from diverse sequencing applications, this article explores an alternative strategy in which we build a common framework and software tool for different applications with similar data structure and analysis goals. Compared to a case-by-case approach, developing a common data analysis framework can offer multiple advantages. A common framework implemented in a general purpose software tool can provide a ready-to-use pipeline for data generated from new applications. It enables one to analyse data from new applications in a timely fashion without developing the whole analysis pipeline from scratch. This can greatly help many researchers to accelerate their research progress. A common framework can also save substantial amounts of developers time and resource. The saved time and resource could be redirected to solving other important problems. When developing a common framework for different applications, however, it is important keep the framework flexible to allow users leverage application-specific data characteristics to ensure most effective use of data. It is also important to implement

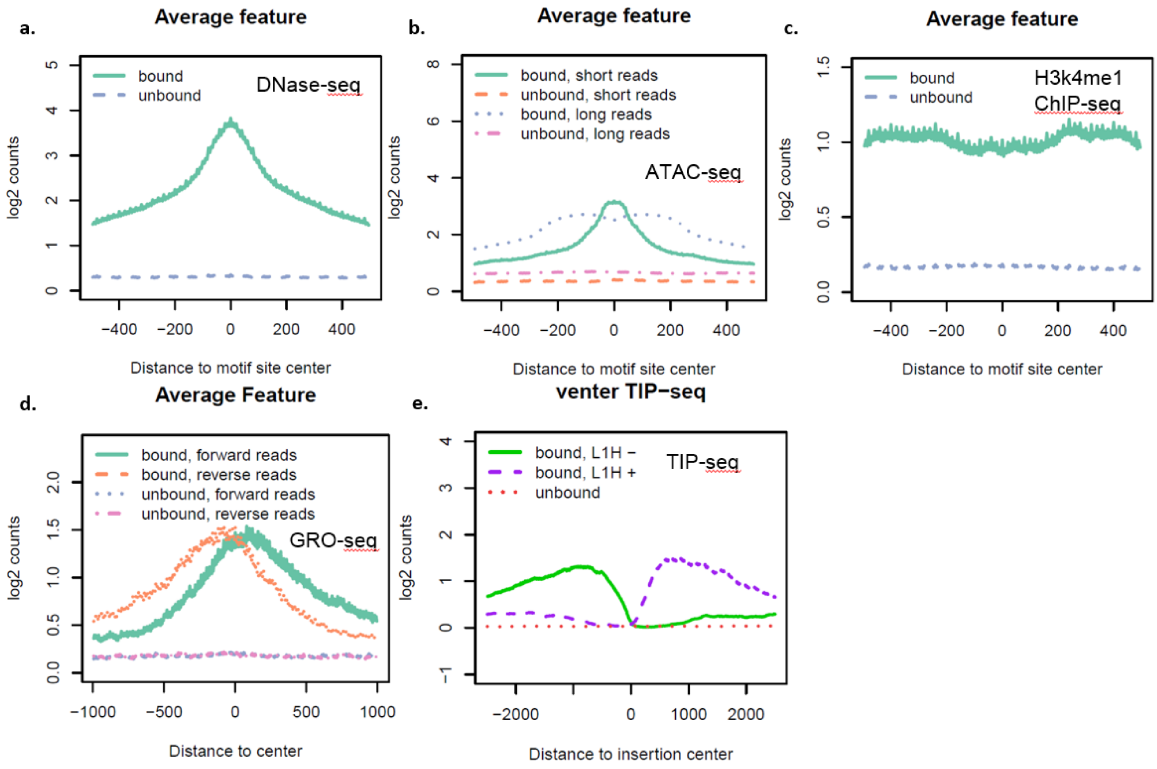

Figure 1.1: Average patterns of features around positive and negative training genomic locations for five applications presented with DNase-seq, ATAC-seq, Histone ChIP-seq, GRO-Seq and TIP-Seq

the framework in a way that can be robustly used in different applications.

The objective of this study is to develop a common and adaptable analysis framework and software tool for a large class of HTS applications where the goal of data analysis is to locate interesting biological signals in the genome based on the signals enrichment and shape features. Examples of applications in this class include mapping transcription factor binding sites using DNase-seq, ATAC-seq and ChIP-seq, detecting active enhancers using GRO-seq, and detecting retrotransposons using TIP-seq, etc. A common feature of all these applications is that signals are indicated by enrichment of aligned sequence reads, and the spatial distribution of reads often show

a characteristic shape. For instance, histone modification ChIP-seq reads around transcription factor binding sites (TFBSs) often display a bimodal peak shape due to positions of flanking nucleosomes surrounding TFBSs(Figure 1.1 c). Reads from TIP-seq for detecting retrotransposons are asymmetrically distributed on one side of each transposon element because one side of the amplified DNA fragments are repetitive sequences in the genome that cannot be aligned (Figure 1.1 e). Analyses of data from these applications can all be handled by building a model that incorporates the signal enrichment and shape information to discriminate signals and noises. However, different applications have different signal enrichment and shape features. Thus, effective use of data requires development of application-specific signal detection models.

In the present study, we focus on a large class of high-throughput sequencing applications where the primary goal of data analysis is to locate interesting biological signals in the genome based on the signals enrichment, shape and other features. Examples of applications in this class include mapping transcription factor binding sites using DNase-seq, ATAC-seq and ChIP-seq, detecting active enhancers using GRO-seq, and detecting retrotransposons using TIP-seq, etc. We develop FCAT, a flexible classification framework and toolbox to deal with this signal detection problem. FCAT has a data pre-processing pipeline that takes aligned sequence reads (BAM files) as input. It saves users time to prepare data for analyses. A supervised learning approach is employed to automatically extract application-specific features that distinguish signals from noises. This allows users to conveniently tailor FCAT to

different applications and at the same time maintain high signal detection power by taking advantage of application-specific data characteristics. In FCAT, users can aggregate multiple learning algorithms including random forests, L1- and L2-regularized logistic regression to improve prediction accuracy and robustness. Furthermore, a non-parametric inference method is developed for estimating false discovery rate of prediction results. We will demonstrate FCAT using multiple applications involving different data types including DNase-seq, ATAC-seq, ChIP-seq, GRO-seq and TIP-seq. These examples show that FCAT offers flexibility and convenience to handle data from different high-throughput sequencing applications. At the same time, it also yields competitive or better signal detection accuracy compared to existing tools for each application. It allows one to greatly increase the efficiency and reduce the burden for developing bioinformatics solutions to the growing number of new sequencing applications.

1.2 Related research

1.2.1 Signal detection in DNase-seq

DNase-seq conducts genome-wide sequencing of hypersensitive genome regions to cleavage by DNase I endonuclease.[4] DNase-seq employs the DNase I enzyme to preferentially digest open chromatin regionsregions that are nucleosome-depletedto reveal gene-regulatory activities.[26] DNase-seq protocol takes advantage of the high-

throughput sequencing technology in traditional assays for DNase I hypersensitive sites (DHS).[27–29] In DNase-seq, the 5 end of the sequence tag marks a DNase I digestion activity. The enrichment and footprints of DNase-seq signals are identified as DHS sites, which are often associated with binding activities of multiple transcription factors.[4]

Computational methods and tools have been developed for predicting transcription factor binding sites (TFBS) from DNase-seq data, including Hotspot,[28,30] F-seq,[31] ZINBA,[32] MACS,[8] CENTIPEDE[21–23] and PIQ.[24] Hotspot is adopted by ENCODE project for the analysis of DHS. Hotspot measures the enrichment of sequence tags by a Z-score, using binomial distribution as the null distribution. In order to capture peaks with relatively weak signals, Hotspot takes a two-phase procedure. In the first phase, hotspot region with high enrichment are identified and the sequence tags that fell in the region are removed in the second phase. Hotspot identifies weaker and reproducible peaks with the highly-enriched regions removed. Two sets of hotspot regions from two phases are combined together as the final result. To calculate the False Discovery Rate, Hotspot makes an assumption that the with no hotspot regions the sequence tags is uniformly distributed and p values are calculated against the uniform distribution.[28,30]

F-seq targets at solving the boundary effects associated with measuring tag enrichment in equal-sized bins used in histogram based-peak calling algorithms. It uses a bandwidth and a Gaussian kernel density function centered at each sequence tag.

F-seq is used for both DNase-seq and ChIP-seq peak calling.[31] ZINBA[32] and MACS[8] were discussed in the Subsection 1.2.2. ZINBA and MACS are also used for peak detection in DNase-seq as well as other high-throughput sequencing assays.

CENTIPEDE is used to detect transcription factor binding sites (TFBS) from DNase-seq.[21] It uses a Bayesian hierarchical framework that develops the prior probability of binding activities based on motif mapping, conservation score as well as other existing knowledge with a logistic regression and uses two different negative binomial distributions conditional on bound and unbound status of the given site. CENTIPEDE can be used in DNase-seq, histone modification ChIP-seq as well as other assays with similar characteristics. The advantage of CENTIPEDE is that it incorporates cell-independent knowledge into prior probability and integrates the knowledge with information from cell-specific experiments in conditional distributions.

PIQ is short for protein interaction quantitation and it is developed to discover transcription factor binding by modelling both magnitude and shape of DNase profile.[24] PIQ uses three steps to estimate TF binding. The first step maps position-weighted matrix from JASPAR, UniPROBE and TRANSFACT genome-wide and identifies computationally mapped motif sites. Similar as CENTIPEDE, the mapping score is incorporated into prior probability of TF binding. In a second step, PIQ performs smoothing of the raw reads with a Gaussian process that adaptively uses reads from neighboring locations to reduce noise. In the final step, expectation

propagation is used to iteratively update the probability of TF binding using information from the magnitude of signals and the shape of footprints. In this study, CENTIPEDE and PIQ are used to benchmark FCAT in related applications.

1.2.2 Signal detection in ChIP-seq

ChIP-seq combines chromatin immunoprecipitation and high-throughput sequencing, which allows researchers to study DNA-protein interactions on a genome-wide scale. Histone modification ChIP-seq gives a comprehensive picture of chromatin packaging.[33] In chromatin immunoprecipitation, antibodies are used to select specific proteins or nucleosomes, which enriches for DNA fragments that are bound to these proteins or nucleosomes. ChIP-seq is the most commonly used assay for profiling binding sites locations for individual proteins and histone modifications. A variety of tools have been developed for peak calling in ChIP-seq data .[8–11,13–20] F-seq discussed in the Subsection 1.2.1 is also used for peak calling in ChIP-seq.[31]

QuEST (Quantitative Enrichment of Sequent Tags) is a statistical framework that uses kernel density estimation for peak calling in ChIP-seq.[9] QuEST first estimates peak shiftdefined as half of the distance between forward and reverse profiles of sequence tags. The forward and reverse profiles are then shifted and combined to give the complete density profile. With peak shift QuEST provides a better estimate for the location of TF binding event. With the combined density profile from forward and reverse tags, QuEST searches for locations with enrichment with the combined

density profile compared to control data.

CisGenome is a software system for assorted analysis with ChIP-seq data. It provides one-and two-sample peak detection algorithms. CisGenome uses a sliding-window approach. In one-sample peak detection, CisGenome identifies regions with large read counts compared against a null distribution with the nonbinding regions. One feature of CisGenome peak calling algorithms is that it uses the negative binomial model instead of the widely-used Poisson model for the background signal. In two-sample peak detection, CisGenome identifies regions with enriched sequence tags compared to those in the control sample using a binomial model.[10]

SISSRs (Site Identification from Short Sequence Reads) takes advantage of directionality of reads, length of fragments and a background model for peak calling in ChIP-seq. SISSRs partitions the genome into nonoverlapping windows with equal sizes and counts reads in each window. It assumes that if the direction of the majority of reads changes to the opposite strand, a binding event is likely to occur. FDR is calculated based on the control sample and a Poisson background model.[11]

FindPeaks[14] is a software system designed for peak calling and extended functionalities for ChIP-seq, including sub-peak identification which separates multiple peaks among a group of overlapping sequence tags, peak-trimming and accommodates different distributions for fragment length distributions. FDR is calculated for each binding site with a Monte Carlo simulation, which compares the number of peaks identified in randomized data with those in real data. PeakSeq exploits the

control sample of ChIP-seq for information on open chromatin. In first pass, PeakSeq identifies putative binding sites and the second phase filters out false positive sites compared to normalized control.[15]

ZINBA (Zero-Inflated Negative Binomial Algorithm) can be used to call broad/narrow peaks in ChIP-seq, DNase-seq and other similar assays. ZINBA models the signals in genomic regions by a zero-inflated negative binomial distribution, which consists of three general components: background regions, enriched regions and regions with artificial zero read counts. Covariate can be incorporated into modeling of the signals. At the end, neighboring enriched regions were merged and boundaries were determined.[32]

MACS is a popular piece of software for calling peaks from ChIP-seq.[8] It is also widely used for peak calling in other epigenomic assays, like DNase-seq and ATAC-seq. MACS models the shift between reads mapped to different strands and uses a Poisson distribution with varying value for the parameter to characterize the background model. FDR is estimated by dividing the number of peaks called in control sample by those in the ChIP sample.

More recently, Ghandi et al. (2014) proposed sequence classifier gapped k-mer support vector machine (gkm-SVM) that uses counts gapped k-mers as feature sets with application in prediction transcription factor binding sites from ChIP-seq and tissue-specific enhancers.[20] Arvey et al. (2012) introduced a support vector machine classifier with k-mer patterns to predict TFBS in ChIP-seq.[18] Agius et al. (2010)

trained a support vector regression classifier that takes advantage of both in vitro protein binding microarray and in vivo ChIP-seq to prediction TFBS.[17] Guo, Mahony and Gifford (2012) developed a computational method called GEM that combines binding detection and motif discovery in one generative probabilistic model for ChIP-seq data.[19]

1.2.3 signal detection in ATAC-seq, GRO-seq and TIP-seq

ATAC-seq is an assay for transposase-accessible chromatin which directly transposes sequencing adaptors in vitro into native chromatin with only 500-50,000 cells. ATAC-seq simultaneously profiles open chromatin, DNA-protein binding, individual and chromatin compaction in one assay. Researchers can gain information through both the position of insertion of transposes and the distribution of insert lengths of fragments in ATAC-seq.[5] Many tools developed for ChIP-seq or DNase-seq can be readily applied to ATAC-seq, including ZINBA,[32] MACS[8] and CENTIPEDE.[21] For example, ZINBA was used to call ATAC-seq peaks using window size of 300bp and offset of 75bp.[5]

Global Run-on sequencing (GRO-seq) maps the position, amount and orientation of transcriptionally engaged RNA polymerases genome-wide.[7] In the GRO-seq protocol, nuclear run-on assay was used to extend transcriptionally engaged RNA

polymerases with BrU tags incorporated on the 5 end. Thus, the orientation, besides the origin, of the transcriptionally engaged nascent RNA can be documented through the assay. Danko et al (2015) introduced discriminative regulatory-element detection from GRO-seq (dREG) that uses support vector regression to identify active transcriptional regulatory elements from GRO-seq data.[25] dREG is a supervised learning algorithm. It uses GRO-cap and high-confidence DNase I hypersensitive sites as positive training sites and trains a support vector regression model with precision nuclear run-on assay (PRO-seq). dREG compiles its feature vector through concatenating normalized read counts mapped to different strand and to nonoverlapping windows with different scales; those windows are centered at the genomic location under investigation. Before model training, dREG filters out the genomic locations with very weak signals and consequently considers those filtered-out positions as negative sites. dREG uses logistic function with two parameters to standardize the read count in each window.

TIP-chip is an assay for identifying transposon insertion sites[34] and TIP-seq further advances the TIP-chip protocol by incorporating the high-throughput sequencing technology. Transposon can jump around their host genome through cut-and-paste or copy-and-paste mechanism. Transposons can inactivate genes, affect gene expression and further change genotype. TIP-chip protocol selectively amplifies transposon flanking regions and hybridizes them to the array to profile transposons in a sample. To the best of our knowledge, there are currently no tools developed for TIP-seq.

To conceptually summarize the reviewed methods above developed for high-throughput sequencing assays, we categorize them into two categories: one is model-based methods, the other is data-driven methods. Model-based methods usually make strong assumptions for signal distribution or generation in regions with true signals and with noises, for example, MACS[8] and ZINBA.[32] This category of methods relies heavily on the applicability of their assumptions in the targeted data type. Data-driven methods consist of supervised learning methods, including dREG.[25] This category of methods exploits training data and guide prediction by trained models. Meanwhile many methods can be considered as hybrid methods, like CENTIPEDE and PIQ. In this study, the propose FCAT is a supervised learning framework. It extracts training data from previously accumulated knowledge and learns signal patterns from data. The learned patterns are then used to guide prediction.

Chapter 2

FCAT workflow

In this chapter, I will describe the methods and implementation of FCAT.

2.1 FCAT overview

Figure 2.1 shows the basic workflow of FCAT. Given a set of read alignment files, the objective of FCAT is to detect genomic loci that carry biological signals of interest. FCAT consists of a training module and a prediction module. The training module takes the read alignment files and a list of training genomic loci as input. The training loci are genomic sites for which the presence or absence of signals is completely or partially known. For each training locus, FCAT extracts features such as read coverage and signal shape with user-specified resolution. Using the extracted features and other user-provided custom features, FCAT constructs signal detection

models that will be used to classify any arbitrary genomic loci into signals or noises. Users can use multiple machine learning algorithms including random forests, L1- and L2-regularized logistic regression to train these models. FCAT can combine prediction results from different models through model aggregation to improve prediction accuracy and robustness. In the prediction module, FCAT applies the trained models to the whole genome or user-specified genomic regions to systematically detect signals. It will report locations of signals along with their estimated false discovery rates.

2.2 Alignment Files

FCAT support both bam and bed files. For each alignment file, users can specify whether the data contain single-end or paired-end reads. For paired-end read data, one can reconstruct DNA fragments using paired reads. FCAT allows users to filter reads based on their mapped strands. One can also use DNA fragment size as a filter and extract features using only fragments that fall within a user-specified size range. Supplementary Table 7.1 lists the parameters that can be set for extracting features from the alignment files.

2.3 Training genomic loci

FCAT uses a supervised learning approach to train signal detection models. This approach allows one to conveniently build models that account for application-specific data characteristics. Supervised learning requires training data. For FCAT, the training data consists of a positive training set and a negative training set. The positive training set contains genomic sites that are known to carry signals. The negative training set contains genomic sites for which signals are highly unlikely to occur. For example, in order to detect active regulatory elements in the HepG2 cell line using DNase-seq data, users may compile a list of regulatory elements (e.g., promoters, enhancers, etc.) known to be active in HepG2 and use them as the positive training set. Similarly, a negative training set may be constructed using random genomic sites. After using these data to train prediction models, the trained models can be applied to scan the whole genome to systematically detect active regulatory elements in HepG2 that are previously unknown.

If genomic loci with known signal status are unavailable, the training data may be prepared by collecting genomic loci that are highly likely or highly unlikely to carry signals. For example, FCAT provides a pre-compiled list of "housekeeping" transcription factor binding sites which can be used to train models for detecting active regulatory elements or transcription factor binding sites using ATAC-seq and/or histone modification ChIP-seq data. The housekeeping binding sites were compiled using publicly available transcription factor (TF) ChIP-seq data available in the Encyclo-

pedia of DNA Elements (ENCODE).[35] For several widely-studied TFs, the ENCODE project has generated ChIP-seq data in multiple cell lines. For each of these TFs, we downloaded all available ChIP-seq narrow peak files from the ENCODE (hg19). Each narrow peak file contains the TFs binding sites in one cell line. We also obtained the TFs DNA binding motif from the JASPAR database and mapped the motif to the human genome using CisGenome under its default parameter setting.[10] Motif sites bound by the TF (i.e., covered by the binding peaks) in all narrow peak files were defined as the TFs housekeeping binding sites. For analyzing ATAC-seq and ChIP-seq data, these housekeeping sites can be used as the positive training set. Although the exact binding status of each housekeeping site in a new cell type is unknown, active TF binding events are highly likely to be enriched in these housekeeping sites. This is because ENCODE data suggest that these sites are likely to be bound by their corresponding TF in a cell-type-independent fashion. Currently, FCAT contains housekeeping sites for CMYC (13 cell lines), CTCF (119 cell lines), E2F (7 cell lines), EGR1 (3 cell lines), and GABP (6 cell lines) (see Supplementary Table 7.2 for more details on cell lines). These sites can be pooled together to serve as the positive training set if users want to use ATAC-seq, histone modification ChIP-seq or other similar data to detect active regulatory elements or predict binding sites of other TFs in a new cell type. Using this pooled housekeeping sites as training data, we assume that a new TF tends to have similar signal patterns as CMYC, CTCF, E2F, EGR1 or GABP in a new cell line. Note that in this pooled positive training

set of TFBS, although CMYC, CTCF, E2F, EGR1 or GABP themselves may have different patterns at their own sites, yet the random forest model included in FCAT's model average framework can capture different patterns of positive sites. FCAT also provides a utility script for randomly sampling genomic sites in the background as negative control sites. These sites can serve as the negative training set in the data analyses. The ratio between positive and negative training regions is set to be approximately 1:1 in the data analyses in this study. Users can adjust the ratio by varying the negative control sites with the utility script (for more details, please see `https://github.com/HeBing/fcat`.

There are many other possible ways to prepare the training genomic loci. For instance, for detecting enrichment signals in a dataset, one could perform an initial peak calling using an existing enrichment-based peak caller such as MACS.[8] Peaks reported by the initial peak calling and genomic loci not covered by the peaks can then be used to compile the positive and negative training sets to train prediction models that incorporate both the enrichment and the application-dependent signal shape information. These refined signal detection models can then be used to reanalyze the data to better identify signals.

2.4 Feature Extraction

FCAT uses both signal intensity and shape as features to construct prediction models. For each genomic locus, a W bp flanking window centered at the locus is considered. The number of reads (single-end sequencing data) or DNA fragments (paired-end sequencing data) within the window is counted and used as the intensity feature. The intensity feature is modeled in count-based benchmark model in FCAT and this model contributes to the final integrated prediction from FCAT. In order to extract shape features, FCAT divides the W bp flanking window centered at each genomic locus into w bp nonoverlapping bins. For each alignment file, the number of reads (single-end sequencing data) or DNA fragments (paired-end sequencing data) falling into each bin is counted. For both intensity and shape features, the bin counts are log2 transformed after adding a pseudo-count of 1.

FCAT can take multiple alignment files as input. Features are extracted from each alignment file. Since FCAT allows users to filter reads or DNA fragments based on their strands or fragment size, one can use the same alignment file to extract multiple sets of features by applying different filters. For example, one can extract a group of features using reads aligned to the positive strand of the genome and extract another group of features using reads aligned to the negative strand. Also, one could extract a set of features using 100-150 bp long DNA fragments and another set of features using 150-300 bp long DNA fragments. For genomic locus r, let $\boldsymbol{b}_i^r = (b_{i1}, \ldots, b_{in})$ be the features extracted from alignment file i from n bins from specified resolution.

Then the final feature vector for locus r is $\boldsymbol{x}_r = (\boldsymbol{b}_1^r, \ldots, \boldsymbol{b}_I^r)^T$, which is obtained by concatenating features from all I alignment files.

FCAT can also incorporate custom features into its predictive modeling framework. In FCAT, all the features are numbered by integer in the format of "1:2.456 2:3.0" indicating 1st feature takes the value 2.456 and 2nd feature takes the value 3.0. FCAT contains a utility script that appends custom features to the feature vector with appropriate integer numbering (for more details, please see `https://github.com/HeBing/fcat`). For example, in the histone ChIP-seq application, to incorporate prior ChIP-seq binding status, the utility script would adapt feature vector "1:2.456 2:3.0" to "1:2.456 2:3.0 3:0" where the 3rd feature is the prior ChIP-seq binding feature.

2.5 Model training and aggregation

Using the extracted features from the training genomic loci, FCAT will train signal detection models via supervised learning. The models will be trained using three base learning methods including random forests (RF),[36] L1-regularized (lasso)[37] and L2-regularized (ridge) logistic regression.[38] For penalized regression, models are fitted with a grid of penalty amount. The random forests build B classification trees. By default, $B = 100$ and each tree selects 10% of total number of features to split on. Users have the option to set these parameters to other

values. Let $T_b(\boldsymbol{x}_r)$ (= 0 or 1) be the b-th tree's prediction for genomic locus r. The RF prediction for locus r is given by $\sum_{b=1}^{B} T_b(\boldsymbol{x}_r)/B$. The penalized logistic regression models assume $\text{logit}(\text{P}\{\text{y}_\text{r} = 1\}) = \boldsymbol{x}_\text{r}^\text{T}\boldsymbol{\beta}$. In the L1-regularized logistic regression, the regression coefficient $\boldsymbol{\beta}$ is estimated by solving the optimization problem $\min_{\boldsymbol{\beta}} \left\{|\boldsymbol{\beta}| + C\sum_{r\in TR} \log\left[1+\exp\left\{-y_r\boldsymbol{\beta}^T\boldsymbol{x}_r\right\}\right]\right\}$ where C controls the amount of penalty and TR represents the set of training genomic loci. In the L2-regularized logistic regression, $\boldsymbol{\beta}$ is estimated by solving the optimization problem $\min_{\boldsymbol{\beta}} \left\{\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C\sum_{r\in TR} \log\left[1+\exp\left\{-y_r\boldsymbol{\beta}^T\boldsymbol{x}_r\right\}\right]\right\}$. For both L1- and L2-regression, models are fitted by setting C to a grid of values (default grid = 0.1, 0.01, 0.001), and FCAT provides users with the option to set their own grid.

Each base learning method is applied to train models using the full set of features under all given parameter settings. Additionally, models are also trained by using the intensity feature only. For each fitted model, the prediction mean square error is computed using training data by three-fold cross-validation. The three top performing models with the smallest mean square errors will be identified, and their prediction results will be aggregated through a weighted average to provide the final prediction. The weights are determined by the inverse of each models mean square prediction error. In other words, let $\hat{y}^m$ be the predicted signal probability from model m for a new locus, and MSE_{cv}^m be the cross-validation mean square error for model m in the training data. The model aggregation result is $\hat{y} = \sum_m \omega_m \hat{y}^m / \sum_m \omega_m$ where $\omega_m = 1/MSE_{cv}^m$. Model aggregation is important for obtaining robust prediction

performance when applying FCAT to handle diverse data types and applications, as the performance of each individual learning method may vary from one application to another.[39] FCAT also provide users with options to choose base learning methods and set their own parameter values.

2.6 Prediction and false discovery rate estimation

Once the models are trained, they will be applied to the whole genome or user-specified genomic regions to detect signals. For each genomic locus, features are extracted in the same way as above. FCAT will then use the trained model to score and rank all genomic loci. In order to determine which loci are statistically significant, we estimate FDR by applying the trained FCAT model to a specified number of control genomic loci reserved from the negative training set. These loci are not used for training FCAT models. We reserved 500 negative control sites in the data analyses in this manuscript. Users can vary the number of negative control sites reserved for FDR with FCAT. After applying FCAT to make predictions at these control loci, the empirical distribution of their predicted results is used as the null distribution to compute p-values for the predicted $\hat{y}$'s. The p-values are then converted into FDR using the Benjamini-Hochberg Procedure.[40]

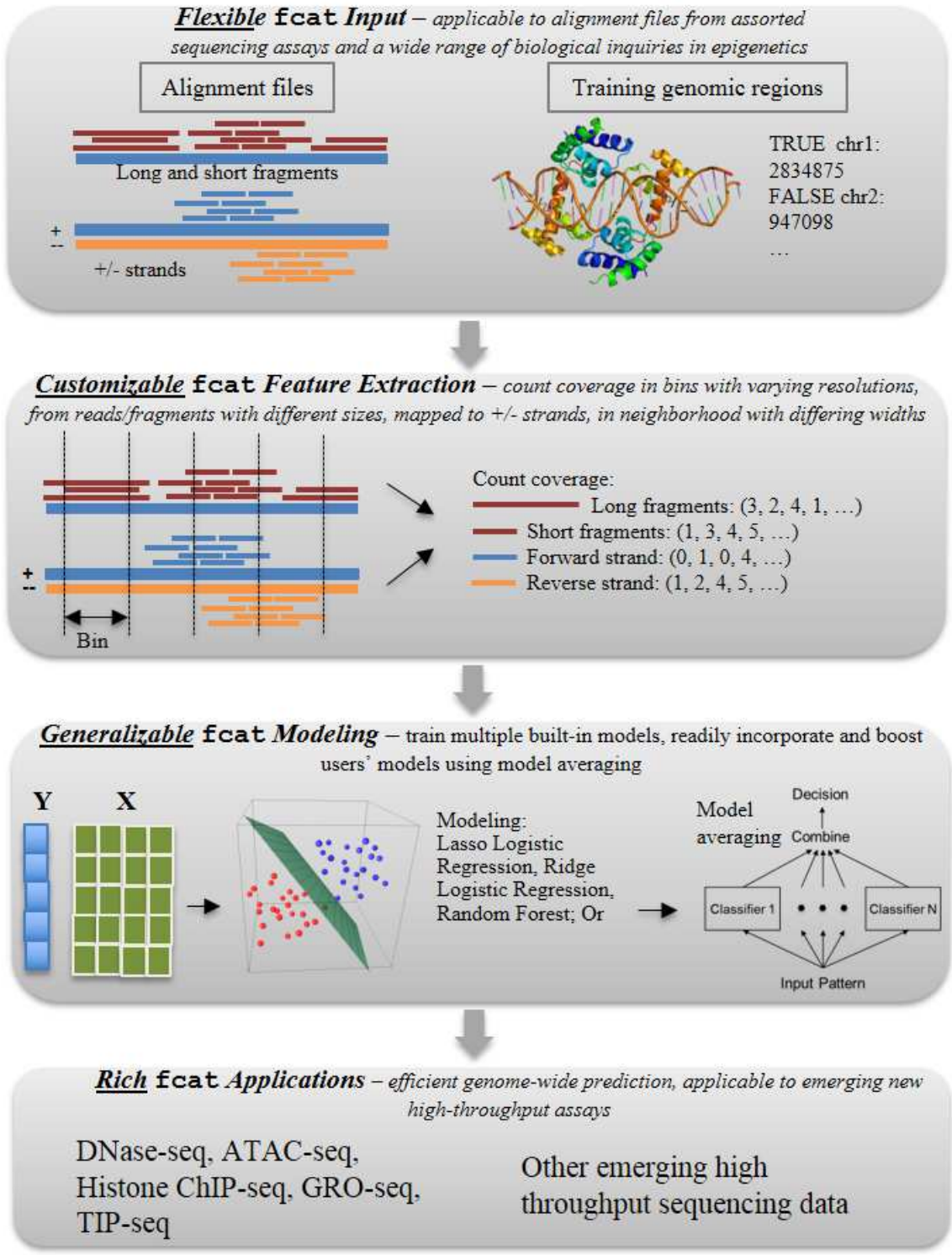

Figure 2.1: An overview of the workflow of FCAT. The input of FCAT includes: a set of alignment files from upstream alignment software and a set of training genomic regions for which we know whether the biological inquiry under investigation exists or not. FCAT first extracts the coverage signals around the training regions as features. FCAT then fits models to features of the training regions. FCAT makes predictions on user-specified genomic sites and combines the prediction results using weighted average from individuals trained models. Finally, FCAT can be applied in a variety of high throughput sequencing data, including DNase-seq, ATAC-seq, histone modification ChIP-seq and GRO-seq as demonstrated here as well as emerging new high throughput sequencing data

Chapter 3

FCAT in predicting transcription factor binding sites (TFBS)

In this chapter, we present three applications of FCAT in predicting transcription factor binding sites with different high-throughput sequencing data type, including DNase-Seq, ATAC-seq and Histone ChIP-seq with combined features.

3.1 Identify transcription factor binding sites from DNase-seq

DNase-seq is a technology for mapping DNase I hypersensitive sites. Since transcription factor binding sites are often marked by DNase I hypersensitivity, one may couple DNase-seq with TF binding motif information to infer TFBSs. FCAT can be

used to handle this task. To demonstrate, we applied FCAT to ENCODE DNase-seq data from two cell lines (Gm12878 and K562) to pre-dict binding sites of a number of TFs. For each test TF and cell line, we first constructed a positive training set by compiling housekeeping binding sites for a number of other TFs using ENCODE ChIP-seq data (see Chapter 2). To ensure that the training and test data are independent, the test TF and cell line were not used for deriving the housekeeping binding sites. We also constructed a negative training set using 4400 randomly sampled genomic sites; the 4900 negative sites include 3900 sites with ratio 1:1 to the number of positive training sites and 500 negative control sites reserved for FDR. For each genomic site, features were extracted from the 1000 bp flanking window for consecutive 5bp bins. To shed light on the discriminating features between signals and noises, Figure 3.1a shows the spatial distributions of DNase-seq reads (i.e., the log2 bin count profile averaged across genomic sites) around the positive and negative training genomic sites in Gm12878. The figure shows that signals around positive training loci display a unimodal shape. The signal gradually decay as one moves away from the locus center. This pattern is not observed for negative training loci.

We used the training data to train FCAT under its default parameter setting. To predict binding sites of a TF, we mapped the TFs motif (Supplementary Table 7.3) to the genome using CisGenome run under its default mode. The trained prediction model was then applied to score and rank each motif site. To evaluate prediction performance, ENCODE ChIP-seq data (narrow peak files downloaded from ENCODE)

for the test TF in the test cell line was used as gold standard (Supplementary Table 7.4). Motif sites covered by ChIP-seq peaks were considered true positives, and the other motif sites were considered true negatives. Using the gold standard, we computed sensitivity (the fraction of true positives that are correctly predicted as TFBSs), specificity (the fraction of true negatives that are correctly predicted as not bound), and FDR (1 the fraction of predicted TFBSs that are true positives). The DNase-seq bam files used for extracting features can be found in Supplementary Table 7.5.

Figure 3.1b and c show the receivers operating characteristics (ROC, i.e., sensitivity vs. 1-specificity) of FCAT for predicting MYC binding sites and E2F binding sites in Gm12878. The ROC curves for other test TFs and cell lines are shown in Supplementary Figures 7.1 and 7.2. A key feature used by FCAT is the signal shape. To demonstrate, FCAT was also run by using the intensity feature only (count-based benchmark). Figure 3.1d demonstrates how signal shape information helped FCAT. The solid blue curve shows the average log2 bin count profile for true positive sites that are correctly predicted by FCAT as signals but incorrectly predicted by the count-based benchmark as noises. The dashed green curve shows the average profile for true negative cases that are correctly predicted by FCAT as noises but incorrectly predicted by count-based benchmark as signals. Clearly, the signal shape helped FCAT to eliminate false positive sites where the total count was high but the signal shape did not match with the pattern observed in the training data (dashed), and it

also increased the sensitivity of FCAT to detect true positive sites where the total count was relatively low but the signal shape was consistent with the shape in the training data (solid).

To further evaluate FCAT, we applied CENTIPEDE and PIQ, two state-of-the-art methods for predicting TFBSs using DNase-seq data to make predictions for the same test TF in the same test cell line. For a fair comparison, we did not incorporate any prior information when running CENTIPEDE (e.g., proximity of a motif site to its nearest transcription start site, evolutionary sequence conservation, etc.). PIQ (version 1.3) was run using its default parameters. Besides CENTIPEDE and PIQ, we also used the popular peak calling algorithm MACS (1.4.0rc 2) to call DNase-seq peaks in each test cell line; and motif sites were ranked by the score of the MACS peaks that covered them.

In Figure 3.1e and f, we showed two specific examples where FCAT helps to filter out false positive sites. There is a motif site at chr1:233749756; based on count-based benchmark model, the count within 1000bp around this site is relatively high (as there are two peaks located at both sides of the motif site) and this site is misclassified as positive based on the count-based benchmark model but is correctly classified as negative by FCAT as the pattern of features, i.e., two partial peaks at both sides of the motif site, is different from what is learned from the training data, i.e., one peak centered at the motif site. The result of FCAT is consistent with the CTCF ChIP-seq data in Gm12878. A similar example is shown for Chr7:93674730 in Figure 3.1f.

Figure 3.1g compares the area under the ROC curves (auROC) of different methods in all test cases, and Figure 3.1h shows the average rank of different methods based on the auROC (a larger value corresponds to a better rank). FCAT provided the best performance in this evaluation. Supplementary Figures 7.3, 7.5, 7.4 and 7.6 further shows each methods sensitivity versus its true FDR. In this evaluation, PIQ performed slightly better than FCAT, and both methods outperformed CENTIPEDE and MACS. Overall, in this well-established application, FCAT demonstrated a performance comparable to PIQ, a method specially designed and optimized for this task, and it outperformed CENTIPEDE and MACS.

3.2 Combine Histone Modification ChIP-seq and covariates to predict TFBS

FCAT can be configured to combine the signal enrichment and shape features with user-provided custom features. For example, consider predicting TFBSs by combining histone modification ChIP-seq and historical TF binding data. TFBSs are often marked by certain types of histone modifications. Similar to DNase-seq, one may predict TFBSs by coupling histone modification ChIP-seq with DNA motif information. This is useful if one wants to use histone modification data in a new cell type to infer binding sites of many TFs simultaneously without conducting many TF ChIP-seq experiments in this cell type. For many TFs, historical ChIP-seq data from

other cell types may be available in public data-bases. These data can provide strong prior information for predict-ing TFBSs in a new cell type. If a motif site is bound by a TF in at least one historical data set, it suggests that the TF is capable of binding to that motif site. Therefore, the motif site would be more likely to be bound by the TF in a new cell type compared to a motif site not bound by the TF in any historical data set. FCAT allows one to conveniently incorporate this prior information when build-ing TFBS prediction models based on histone modification ChIP-seq.

To demonstrate, we applied FCAT to predict TFBS using H3K4me1 ChIP-seq in Gm12878. H3K4me1 is one commonly studied type of histone modification, representing the monomethylation of the 4th residue from the start of the H3 protein. The training genomic loci and the intensity and shape features are prepared in the same way as Subsection 3.1 where DNase-seq was used for prediction. To prepare the historical prior, for each TF we collected all available ENCODE ChIP-seq data (narrow peak files) for that TF (see Supplementary Table 7.6). After excluding ChIP-seq data from the test cell line, a feature that encodes historical information is computed for each motif site. For motif sites bound by the TF (i.e., overlap with a ChIP-seq peak) in any cell line, the feature value was set to 1. For motif sites not bound by the TF in any cell line, the feature value was set to 0. FCAT was then trained using both the features extracted from the histone modification ChIP-seq and features from the historical TF ChIP-seq data. The histone modification ChIP-seq BAM files used for extracting features can be found in Supplementary Table 7.5.

Figure 3.2 a,b,c and Supplementary Figure 7.16 show the ROC curves for FCAT with only H3K4me1 features versus FCAT with both H3K4me1 and historical features for different TFs in the Gm12878 cell line. FCAT with the combined features clearly out-performed FCAT with only H3K4me1 features. FCAT with only H3K4me1 features in turn outperformed FCAT that use intensity features only. We further compared FCAT with CENTIPEDE, PIQ and MACS. Since historical information cannot be used as input for these methods, CENTIPEDE, PIQ and MACS analyses are performed using only H3K4me1 data. Figure 3.2d compares the auROC of different methods, and Figure 3.2e shows the average rank of each method. FCAT that combines H4K3me1 and historical information showed the best performance. Supplementary Figures 7.17 and 7.18 further compares different methods using sensitivity versus FDR curves. The conclusions were similar. This example demonstrates the flexibility of FCAT. Here the ability to conveniently customize the analysis by incorporating historical information has played a key role in improving the signal detection.

3.3 Infer transcriptional factor binding sites from ATAC-seq

ATAC-seq is another technology for mapping active regulatory elements. By coupling ATAC-seq data with DNA motif sites, one can predict TFBSs. FCAT can also

be used to handle this task. To demonstrate, we applied FCAT to ATAC-seq data from the Gm12878 and K562 to predict binding sites of a number of TFs. For each test TF and cell line, we first constructed a positive training set by compiling housekeeping binding sites for a number of other TFs using ENCODE ChIP-seq data (see Chapter 2). To ensure that the training and test data are independent, the test TF and cell line were not used for deriving the housekeeping binding sites. We also constructed a negative training set using 4400 randomly sampled genomic sites. To extract features for each genomic site, the paired-end ATAC-seq reads were categorized based on their fragment lengths. Previously, it was shown that short DNA fragments may be used to mark the nucleosome free regions where transacting proteins bind to DNA, and long fragments may be used to identify nucleosome locations. Since both pieces of information may be useful for marking active TFBSs, we used FCAT to extract features from the short fragments (less than 100bp, marking nucleosome free regions) and long fragments (180 - 247 bp, marking mononucleosomes)[5] separately and then concatenated them together. For each fragment size range, features were extracted from the 1000 bp flanking window for consecutive 5-bp bins. Figure 3.3a shows the spatial distributions of ATAC-seq signals around the positive and negative training genomic sites in Gm12878. It shows that signals from nucleosome free fragments are unimodal, marking the TF binding site. Signals from mononucleosome reads, on the other hand, are bimodal, marking the two flanking nucleosomes. This pattern is not observed for negative training loci.

Similar to DNase-seq, we used the training data to train FCAT under its default parameter setting. To predict binding sites of a TF, we mapped the TFs motif (Supplementary Table 7.3) to the genome using CisGenome run under its default mode.[10] The trained prediction model was then applied to score and rank each motif site. To evaluate prediction performance, ENCODE ChIP-seq data (narrow peak files downloaded from ENCODE) for the test TF in the test cell line was used as gold standard (Supplementary Table 7.3). Motif sites covered by ChIP-seq peaks were considered true positives, and the other motif sites were considered true negatives. Using the gold standard, we computed sensitivity (the fraction of true positives that are correctly predicted as TFBSs), specificity (the fraction of true negatives that are correctly predicted as not bound), and FDR (1 the fraction of predicted TFBSs that are true positives). The ATAC-seq bam files used for extracting features can be found in Supplementary Table 7.5.

Figure 3.3b shows the receivers operating characteristics (ROC, i.e., sensitivity vs. 1-specificity) of FCAT for predicting SRF bind-ing sites in Gm12878. The ROC curves for other test TFs and cell lines are shown in Supplementary Figure 7.7. A key feature used by FCAT is the signal shape. To demonstrate, FCAT was also run by using the intensity feature only (count-based benchmark). Figure 3.3c and d demonstrate how signal shape information helped FCAT. The solid blue curves show the average log2 bin count profile for true positive sites that are correctly predicted by FCAT as signals but incorrectly predicted by the count-based benchmark as noises. The dashed green

curve shows the average profile for true negative cases that are correctly predicted by FCAT as noises but incorrectly predicted by count-based benchmark as signals. Clearly, the signal shape helped FCAT to eliminate false positive sites where the total count was high but the signal shape did not match with the pattern observed in the training data (dashed), and it also increased the sensitivity of FCAT to detect true positive sites where the total count was relatively low but the signal shape was consistent with the shape in the training data (solid).

To further evaluate FCAT, we applied CENTIPEDE and PIQ, two state-of-the-art methods for predicting TFBSs using DNase-seq data to make predictions for the same test TF in the same test cell line. For a fair comparison, we did not incorporate any prior information when running CENTIPEDE (e.g., proximity of a motif site to its nearest transcription start site, evolutionary sequence conservation, etc.). PIQ (version 1.3) was run using its default parameters. Besides CENTIPEDE and PIQ, we also used the popular peak calling algorithm MACS (1.4.0rc 2) to call DNase-seq peaks in each test cell line; and motif sites were ranked by the score of the MACS peaks that covered them.

Figure 3.3e further provides a specific example where FCAT accurately predicts binding activity while count-based benchmark model missed it. It is seen that for chr11:106027113 the average count around this motif site is quite low and thus is labelled as negative by count-based benchmark model. However, these fragments, though only a handful, forms a unimodal pattern with short fragments and peaks

around the site based with the long fragments; thus it is correctly recognized by FCAT. This binding site is further confirmed by ENCODE Gm12878 TFBS uniform peaks of CTCF.

Figure 3.3f compares the area under the ROC curves (auROC) of different methods in all test cases, and Figure 3.3g shows the average rank of different methods based on the auROC (a larger value corresponds to a better rank). Supplementary Figures 7.10 and 7.12 presents auAUC of predictions results for K562. Supplementary Figures 7.8, 7.11, 7.9 and 7.13 further shows each methods sensitivity versus its true FDR. In these evaluations, FCAT provided the best performance.

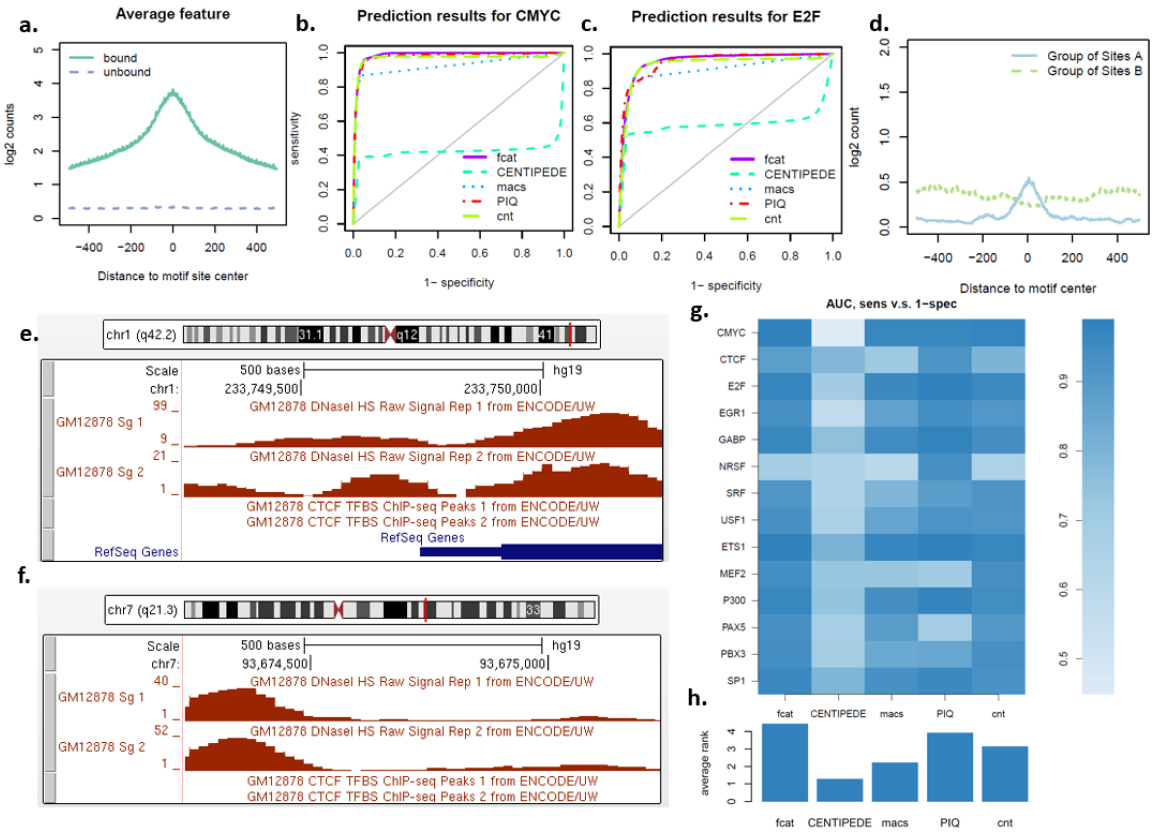

Figure 3.1: FCAT results for identifying TFBS from DNase-seq. (a) average DNase-seq signals around bound and unbound training sites. (b) and (c) FCAT sensitivity versus MACS, CENTIPEDE and PIQ for CMYC and E2F, respectively. (d) DNase-seq signals for true positive cases (i.e., Group of sites A) and true negative cases (i.e., Group of sites B) where FCAT made the correct prediction while the count-based benchmark model yielded the wrong prediction. (e) A concrete example of a CTCF motif site at chr1: 233749756. It is not active according to GM12878 CTCF ChIP-seq. Based on the count-based benchmark, the relatively high read count around the site yielded a high probability of being positive. However, FCAT correctly recognized that the bimodal shape does not match with the learned pattern and gives a very low probability of it being active. (f) Another example at chr7:93674730. It is not active according to CTCF ChIP-seq in Gm12878. The peak at the left side of the motif site increased the read count in the window and caused the benchmark model to falsely predict it to be an active site. FCAT did not see the learned pattern at the motif site and correctly filtered out this false positive. (g) Heatmap for area under curves for ROC curves for prediction results of multiple TFs in Gm12878. (h) shows the barplots of average ranks of performance among FCAT, CENTIPEDE, MACS and PIQ for Gm12878.

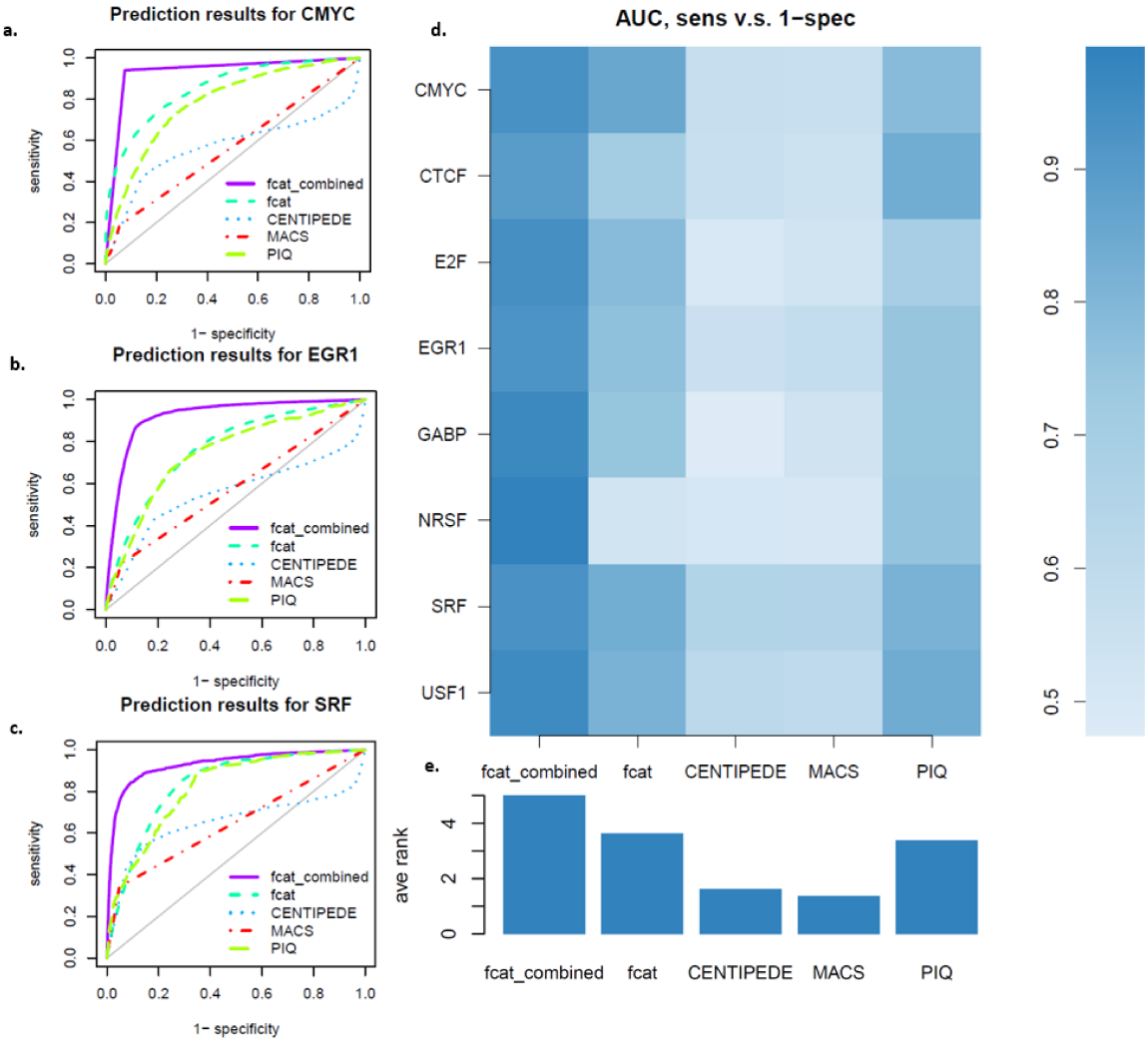

Figure 3.2: FCAT results for identifying TFBS from histone modification H3K4me1 and combined features of H3K4me1 plus historical information. (a) (b) (c) FCAT ROC curves comparing FCAT with H3K4me1 and combined features for CMYC, CTCF and SRF, respectively; ROC curves for CENTIPEDE, MACS and PIQ were obtained using H3K4me1 only. (d) Heatmap for area under curves for ROC curves for prediction results of multiple TFs in Gm12878. (e) shows the barplots of average ranks of performance comparing FCAT with H3K4me1 only and FCAT with combined features as well as CENTIPEDE, MACS and PIQ.

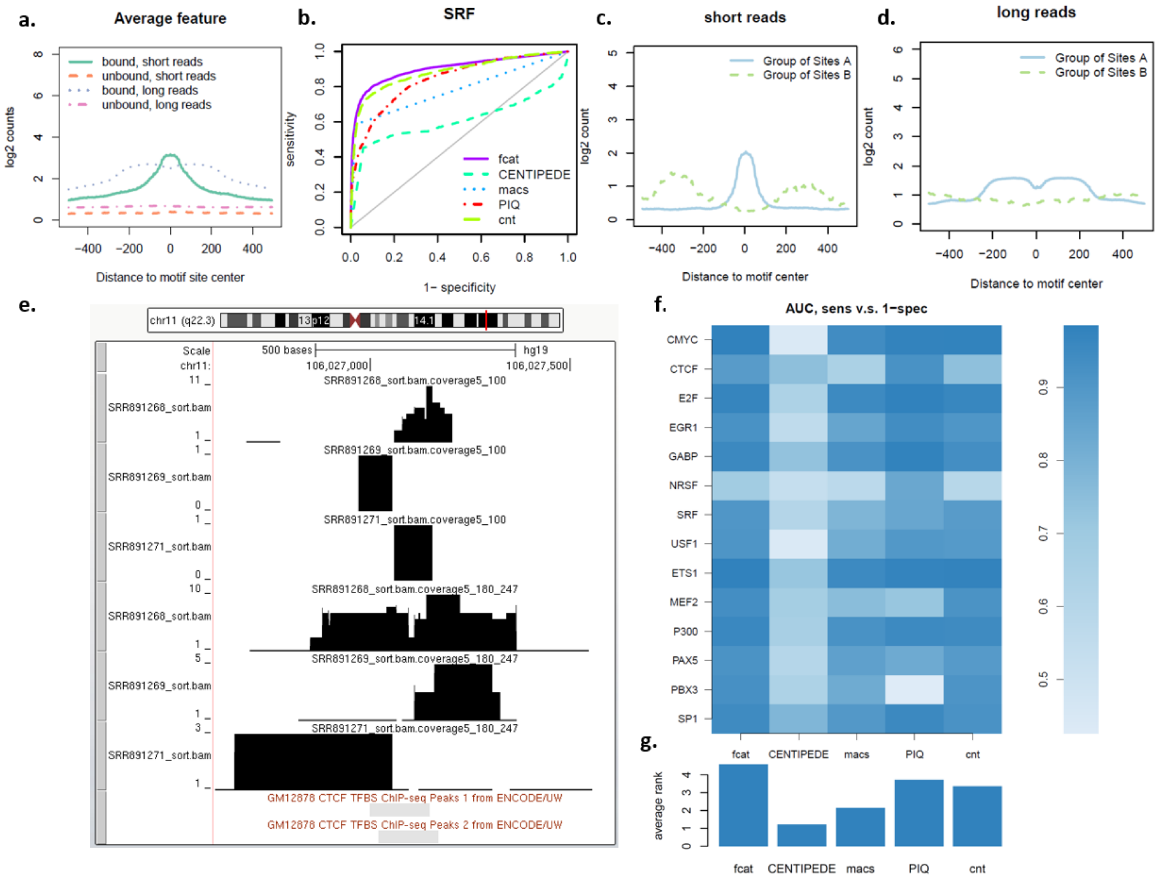

Figure 3.3: FCAT results for identifying TFBS from ATAC-seq. (a) average ATAC-seq signals around training sites. (b) FCAT sensitivity versus MACS, CENTIPEDE and PIQ for SRF in Gm12878. (c) and (d) ATAC-seq signals for true positive cases (i.e., Group of sites A) and true negative cases (i.e., Group of sites B) where FCAT made the correct prediction while the benchmark model yielded the wrong prediction for short fragments (left) and right fragments (right), respectively. (e) An example of a motif site at chr11:106027113; based on the benchmark model, the average read count around this site is relatively low and this site is misclassified as negative but is correctly classified as positive by FCAT as the pattern of features shows a small peak centered at the motif site for short fragments and peaks around the site for long fragments. This FCAT-detected site is confirmed by CTCF ChIP-seq Gm12878 from ENCODE/UW. (f) Heatmap for area under curves for ROC curves for multiple TFs including CMYC, CTCF, E2F, EGR1, GABP, NRSF, SRF, USF1, ETS1, MEF2, P300, PAX5, PBX3, SP1 using ATAC-seq in Gm12878. (g) barplot for the average ranks for area under curves for FCAT, CENTIPEDE, MACS and PIQ across different TFs.

Chapter 4

FCAT in predicting the status of known enhancer RNA with GRO-seq

In this chapter, we present an application of FCAT in predicting whether known enhancer RNA are active or not in a new cell from GRO-seq.

4.1 Detect enhancer RNA from strand-specific footprint using GRO-seq

GRO-seq is a technology designed to locate and quantify the amount and orientation of nuclear run-on RNA polymerase genome-wide. Data from GRO-seq provide a

snapshot of nascent RNA molecules associated with transcriptionally engaged polymerase.[7] Previous studies showed that many enhancer RNA (eRNA) are transcribed on both the forward and reverse DNA strands.[41] This bi-directional transcription may serve as a signature to detect active enhancers using GRO-seq data. Below we demonstrate that FCAT can also be used for this task.

We configured FCAT to extract features separately from the forward-stand and reverse-stand reads and then concatenated these two sets of features as the feature vector for FCAT. To compile the positive training loci, we downloaded 65,423 human permissive enhancers from Fantom5[42,43] and further filtered them by eliminating those not covered by the DNase I hypersensitive sites of the available ENCODE Duke and UW uniform DNase-seq narrow peaks (14 cell lines, please see Supplementary Table 7.7 for more details). We obtained 926 positive training cases for Gm12878 and 879 for K562. To ensure that the training and test data are independent, for each test cell type the positive training set was constructed by excluding the test cell type from the filter. We also randomly sampled 1000 genomic loci and used them as negative training set for Gm12878 and K562, respectively. Using the training data, we trained FCAT under its default parameter setting. We then applied it to the list of Fantom5 enhancers to predict whether those known enhancers are active or not in Gm12878 and K562 respectively. For evaluation, Fantom5 enhancers that overlap with the ENCODE DNase I hypersensitive sites in each test cell type were identified and used as gold standard.

Figure 4.1a shows the GRO-seq signal around positive training enhancers in K562 GRO-seq produces one peak around the train-ing loci from each strand. Figure 4.1b and c show the ROC curves for K562 and Gm12878, respectively. Figure 4.1d provides intuition for the performance of FCAT. It shows the average signal patterns for true positive sites (i.e., Group A) and true negative sites (i.e., Group B) that were correctly predicted by FCAT but incorrectly predicted by the count-based benchmark. This figure shows that FCAT was able to eliminate false positive sites with high read count but no characteristic bi-directional signal shape that cannot be eliminated by the count-based benchmark. It was also more sensitive for detecting true positive sites with low read counts but showing the characteristic bi-directional signal shape which cannot be detected by the count-based benchmark. Figure 4.1e shows a concrete example at chr3:14413456. Based on benchmark model, the count around this site is quite low and this site is misclassified as negative but is correctly classified as positive by FCAT as the features show one peak at the forward strand and one at the reverse strand. This FCAT-detected enhancer RNA site is confirmed by Gm12878 DNase hypertensive site, Gm12878 H3K4me1 and Gm12878 H3K27ac markers from ENCODE. Figure 4.1f give another example for chr1:53107133.

We further compared FCAT with MACS and dREG.[25] dREG is a software specifically developed for identifying active transcriptional regulatory elements from GRO-seq using support vector regression. Figure 4.1b and c show that FCAT performed better than MACS and for some parts outperformed dREG. Figure 4.1g and h com-

pares the area under the ROC of different methods. FCAT showed the best performance. Supplementary Figures 7.14 and 7.15 further showed comparison based on the sensitivity versus true FDR curves.

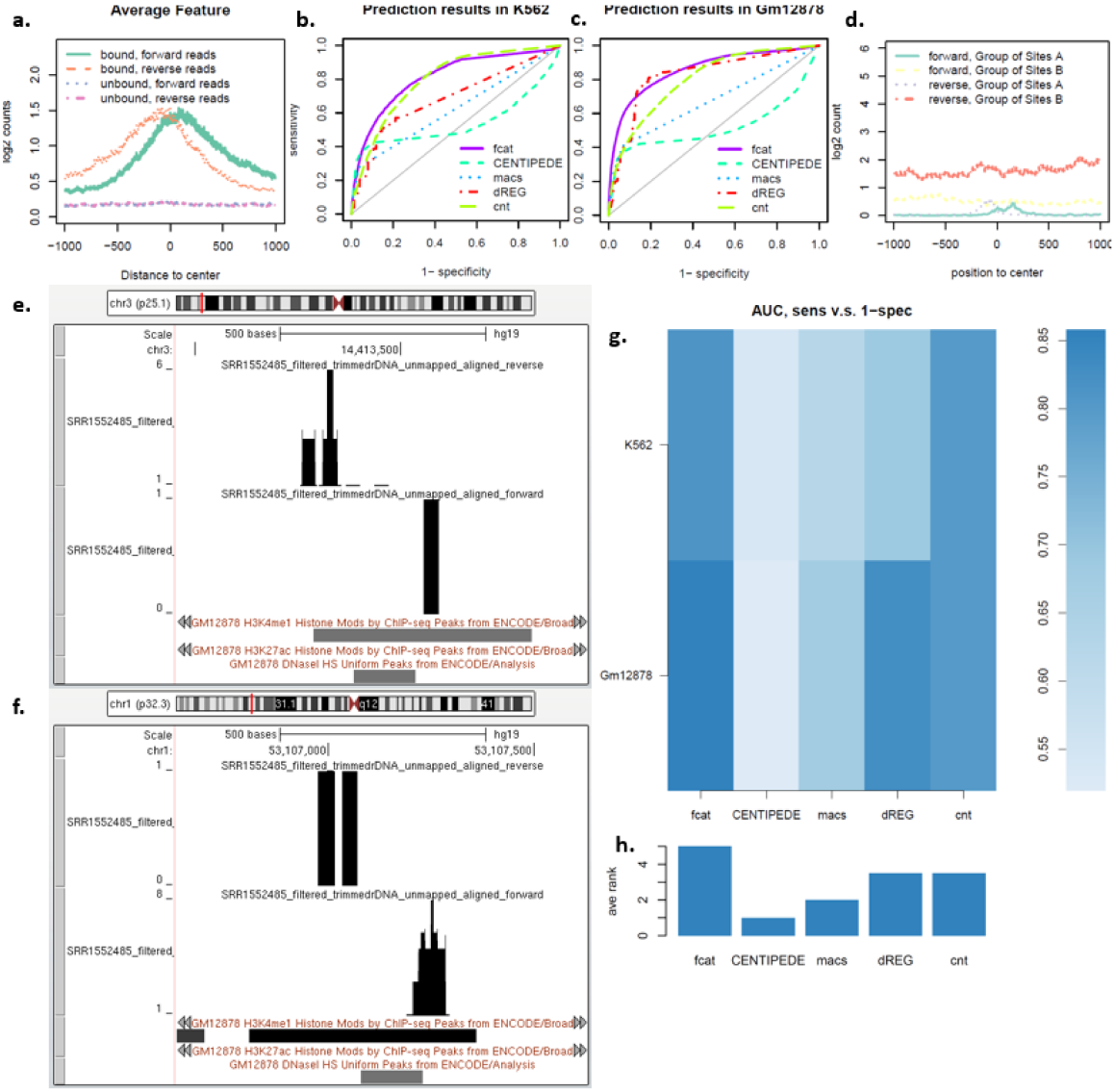

Figure 4.1: FCAT results for identifying eRNA from GRO-seq. (a) GRO-seq signals around training sites for forward and reverse stand for bound and unbound training regions. (b) and (c) FCAT sensitivity versus MACS, CENTIPEDE and dREG in K562 and Gm12878, respectively. (b) GRO-seq signals for true positive cases (i.e., Group of sites A) and true negative cases (i.e., Group of sites B) where FCAT made the correct prediction while the benchmark model yielded the wrong prediction. (e) A concrete example at chr3:14413456. Based on benchmark model, the average around this site is quite low and this site is misclassified as negative but is correctly classified as positive by FCAT as the features show one peak at the forward stand and one at the reverse strand. This FCAT-detected enhancer RNA site is confirmed Gm12878 DNase hypersensitive site, Gm12878 H3K4me1 and Gm12878 H3K27ac markers. (f) chr1:53107133 is another example in which the forward and reverse reads form peaks on different side of the candidate site and is correctly recognized as positive sites by FCAT. However, due to the relatively low count of reads, the benchmark misclassified the sites as negative. (g) presents the heatmap for AUC of prediction performance comparing FCAT to MACS, CENTIPEDE, and dREG; (h) shows the average rank of AUC for the four algorithms.

Chapter 5

FCAT in emerging high-throughput data TIP-seq

In this chapter, we present an application of FCAT in an emerging high-throughput sequencing data TIP-seq, which profiles the transposon insertion sites genome-wide.

5.1 Identify transposon insertion sites from TIP-seq

The two examples above evaluate FCAT based on well-established applications. We used those applications to demonstrate that FCAT can offer competitive performance compared to methods designed specifically for each application. FCAT, however, is most useful when applied to new HTS applications for which good data

analy-sis solutions are still lacking. Below we will illustrate this via two examples.

Transposable elements (i.e., transposons) occupies a substantial fraction of human genome.[44] L1 retrotranspos-ons is a class of transposable elements that can still actively jump-ing in the human genome to affect phenotypes.[45,46] TIP-chip is a recent technology for systematically mapping L1 retrotransposons including their de novo insertion sites in human disease samples.[34,47,48] The sequence of L1 is highly repetitive in the genome. TIP-chip uses specially designed primers to amplify the highly repetitive L1 retrotransposons together with some non-repetitive flanking sequences at one end of the retrotransposons. Since the non-repetitive flanking sequences can be uniquely mapped in the genome, they can be used to locate L1 retrotransposons. TIP-chip uses specially designed DNA microarrays to detect such non-repetitive flanking sequences. Recently, by using sequencing to replace microarrays, TIP-chip has evolved into TIP-seq. Since TIP-seq is a new technology, analysis tools specifically designed for TIP-seq data are still lacking. Here we show that FCAT can be readily applied to analyzing TIP-seq data.

We obtained TIP-seq data from venter family. TIP-seq is often used to detect de novo L1 insertions. However, it is difficult to compile a validated set of de novo L1 insertions in patient samples for benchmarking. Therefore, for method evaluation purpose, we downloaded 1,550 known LINE-1 human specific (L1-HS) insertion sites in the human reference genome from euL1db.[49] We then randomly partitioned these sites into a positive training set of 1000 sites and a test set of 550 sites. We also

randomly sampled 1500 genomic sites and used them as the negative training set. For each site, features were extracted from 5000 bp flanking window for 5 bp consecutive bins. The window size 5000 bp was chosen based on the typical width of the TIP-seq signal.[34] Since high-quality gold standard for genome-wide profile of transposon insertion is difficult to find, here we apply FCAT on a compiled testing dataset. After training FCAT using the training data, it is applied to a testing dataset including 550 positive sites and 700 random background regions.

Figure 5.1a and b shows the average signal profile around the training genomic sites for L1H insertion towards positive strand and negative strand, respectively. One can see that depending on which strand an L1 element is located, the TIP-seq signal is uni-directional. In other words, only one side of the L1 insertion site has clear high signals. This is because only one end of the amplified DNA fragments can be uniquely mapped to the genome, and the other side are repetitive sequences that cannot be mapped. Figure 5.1c shows the ROC curves. We compared FCAT with MACS and it shows that FCAT outperforms MACS substantially. Figure 5.1d shows the sensitivity versus FDR, and the conclusion was similar.

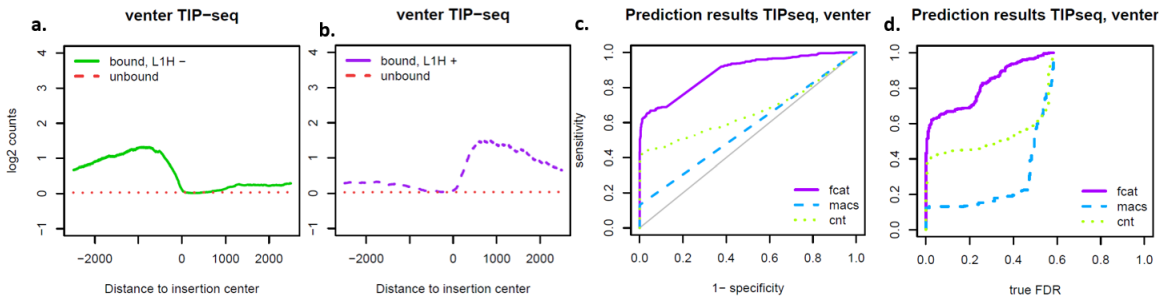

Figure 5.1: FCAT results for identifying L1H insertion sites from TIP-seq. (a) and (b) TIP-seq signals around training sites for L1H. (c) ROC curves for FCAT and MACS for TIP-seq. (d) sensitivity versus true FDR comparing FCAT and MACS

Chapter 6

Discussion and Conclusion

In this chapter, we discuss the conclusion, limitation and future directions of FCAT. To summarize, FCAT offers a general platform that can be conveniently adapted to a variety of different HTS applications to detect signals. The supervised learning approach used by FCAT allows one to build signal detection models that account for application-specific characteristics. This provides the key for making FCAT flexible to different applications without sacrificing the power of signal detection. In fact, we have shown that FCAT offered competitive performance in well-established applications DNase-seq, ATAC-seq, ChIP-seq and outperformed other methods in new applications GRO-seq and TIP-seq.

More specifically, two advantages of FCAT contributes primarily to its flexibility and adaptivity. First, the customizable feature extraction module calculates the counts of reads that satisfy a set of user-defined criteria directly from the BAM

file format. The BAM file format is the industrial standard alignment file format for high-throughput sequencing technology. The ENCODE project uses BAM files for its data production storage.[35] BAM file is a binary file with complex header specifying the meta data for data fields. FCAT can filter out reads/fragments with different lengths, mapped to different strands, mapped to windows with different width and can concatenate vector of counts from different groups of reads to incorporate information from different aspects and at varying scales. For example, in the study, in ATAC-seq application, counts from short fragments and long fragments are calculated separately and concatenated to form the feature vector for ATAC-seq in window centering at motif site with length 2000bp. In GRO-seq, counts from reads mapped to forward and reverse strands were calculated separately and concatenated. In TIP-seq, read counts from a 5000bp window centering at the locus under investigation. For new applications, FCAT can be easily customized to compile tailored feature vectors.

Second, FCAT takes a supervised learning approach at the problem of signal detection that ensembles prediction results from a variety of statistical models to produce the final result. For previously studied research topics like TFBS, compiled training data is provided with FCAT. For new application, FCAT allows researchers to shape the learning process by providing scenario-based training data in a standard format. Meanwhile, instead of using a single model, FCAT incorporates multiple modelsL1 penalized logistic regression, L2 penalized logistic regression and random forestand employs ensemble learning for the prediction. If one uses individual learning

methods without aggregation, it is difficult to make the signal detection performance standing consistently at the top. For instance, the best model for predicting CTCF binding sites in ATAC-seq was random forest, whereas the best model for predicting CMYC binding sites in H3k4me1 with combined feature was L1-regularized logistic regression (Supplementary Figure 7.19). With FCAT, the weighted average of the top-performing models is used as the final results. In predicting CTCF binding sites with ATAC-seq, heavy weight was given to random forest model, whereas in prediction CMYC binding sites with H3k4me1 with combined feature L1-regularized logistic regression was given the highest weight.

Additionally, combining the two advantages discussed above, FCAT can integrate different types of data in the predictive framework, including different types of sequencing assays or sequencing assay with covariates. For example, in the application of histone modification of ChIP-seq data, features from H3k4me1 ChIP-seq and a categorical covariate summarizing historic information were concatenated and thus integrated in FCAT. As it was shown in the results, the added covariate improved the prediction accuracy and successfully augments the value of a single experiment with previous accumulated knowledge. In another scenario, if multiple types of sequencing assays are performed on the same sample, features can be extracted separately from each assay and concatenated together to form a long feature vector for FCAT training and prediction. In this way, researchers can integrate information from different data types in a straight-forward manner.

Allowing users to use customized parameters in FCAT contributes to its flexibility. Among the customizable parameters, we have the width of the flanking window at each training region W and the size of the bin w for feature extraction. The values for W and w in practice can be chosen based on biological and technical considerations. From a biological perspective, the bin size w needs to be small enough to capture the loci of the biological activity of interest. Take prediction of TFBS as an example. In the applications of predicting TFBS presented here, bin size is chosen to be 5bp. The length of known motifs of TFs lies in the range between around 5bp to more than 30bp.[50] The final prediction from FCAT indicates how likely a TF binds with the DNA in a bin. If the bin size is too large, it is not able to detect the specific location for TFBS. Thus when determining bin size w, we need to consider the spacial resolution of the biological activity of interest in practice. At the same time, if w is too small, for example, 1bp, it requires much computing resources to train the models and make prediction. Thus when using FCAT in practice, both biological and technical issues should be considered. The value of W should be chosen depending on specific application. The general rule is that W should be large enough to include the full pattern around training sites. For example, in the TIP-seq application, according to previous literature,[34] the pattern of signals spans a window of around 5000bp and thus W was chosen to be 5000bp in the TIP-seq application. If there exists no prior information about the spacial resolution of the biological activity under investigation or the width of signal patterns, cross validation can be used to determine the values

of W and w. Our experiences with FCAT reveal that it is helpful to use domain knowledge in the application of interest to inform the choices of w and W.

FCAT provides a whole pipeline for extracting features from alignment files with rich options, configuring prediction methods, and making predictions. This pipeline can save valuable time of users when they want to develop custom data analysis solutions to new HTS applications. This can greatly help researchers to effectively use new data from new experiments in a timely manner. Like all other methods, FCAT also has limitations. First, FCAT requires training data. In some new applications, compilation of training data may not be straightforward. However, with training data, the power of supervised learning can make up for the time spent in compiling training data. Additionally, existing data/knowledge about the same biological inquiries can also be used to compile the training data. Second, FCAT currently only supports three base learning methods. Currently, FCAT only supports classification with two classes. It can be easily generalized to classification with multiple classes. In the future, more methods will be added to FCAT to enrich the selection of base models. As we continue to add functions to FCAT, we hope to make it a general and powerful tool for turning HTS data into discoveries.

Driven by technological advances, we have witnessed a deluge of new high-throughput sequencing assays for interrogating different properties of genomes on a genome-wide scale, for example, DNase-seq, ATAC-seq, GRO-seq and TIP-seq. Each of these assays provides a unique, yet complementary, view of the genome. FCAT is an attempt to-

wards a shared, unified statistical and computational framework for high-throughput sequence assays. Currently the applications FCAT can handle primarily lie in the area of epigenomics. Future development of FCAT can be driven towards a shared framework. On one hand, with new application emerging, researchers can share training data used for signal detection in this new application. For example, Hi-C is a high-throughput sequencing assay designed to capture the conformation of genomes.[51] We can extract information from the ligation products formed by covalently-linked DNA fragments to form our features in this new application, which can be further used to reveal the 3D organization of chromatin. On the other hand, new models can be plugged into FCAT to augment the ensemble learning system. A new model with a train and predict method can be easily plugged into FCAT and contributes to the model averaging in FCAT.

Chapter 7

Appendix

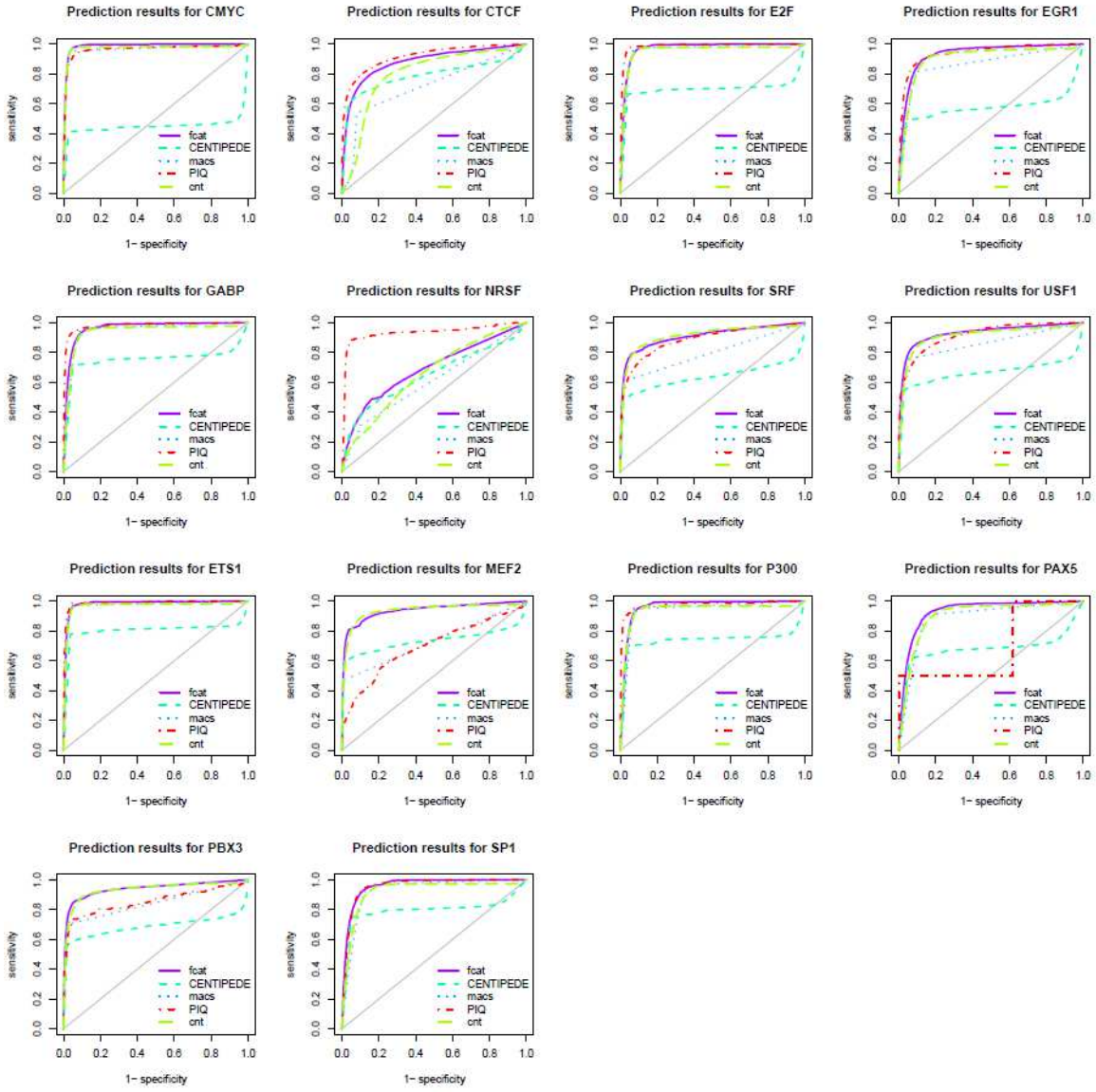

Figure 7.1: Prediction results of ROC for TFBS from DNase-Seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in DNase-Seq, CENTIPEDE, MACS, and PIQ in Gm12878.

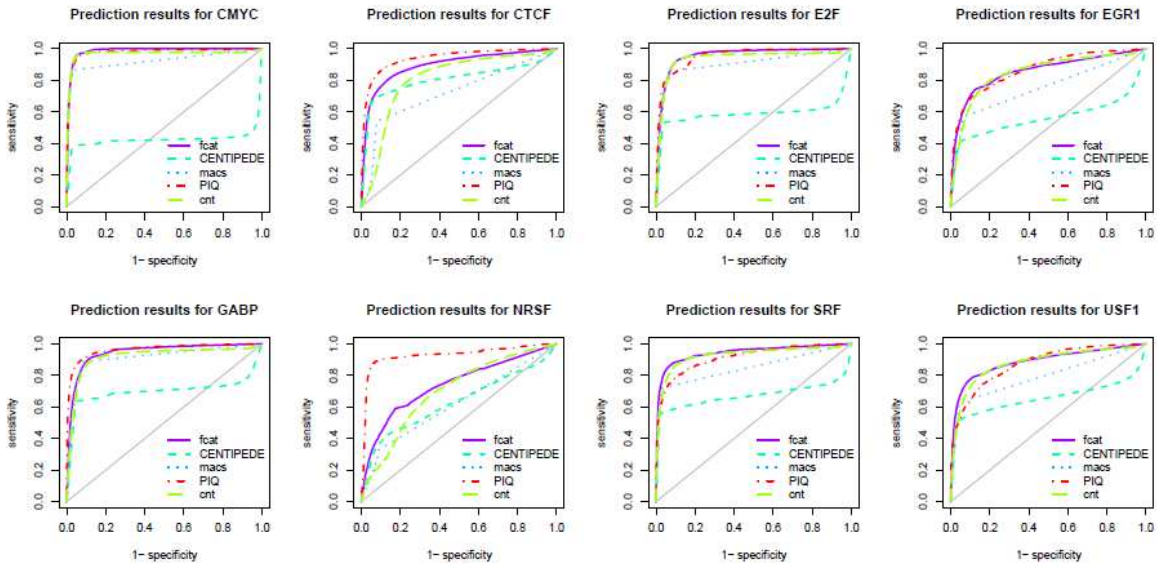

Figure 7.2: Prediction results of ROC for TFBS from DNase-Seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in DNase-Seq, CENTIPEDE, MACS, and PIQ in K562.

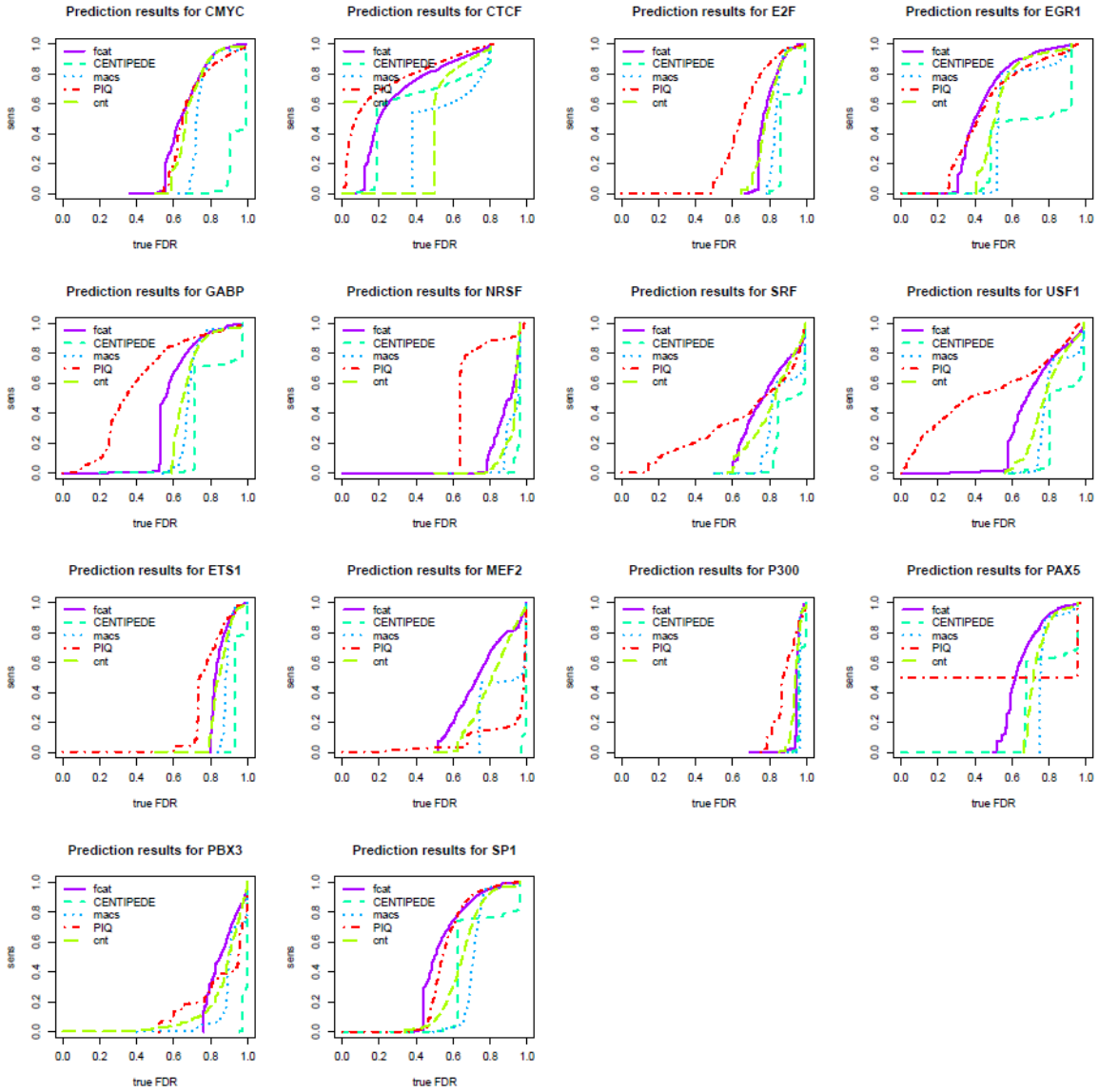

Figure 7.3: Prediction results of sensitivity versus true FDR for TFBS from DNase-Seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in DNase-Seq, CENTIPEDE, MACS, and PIQ in Gm12878.

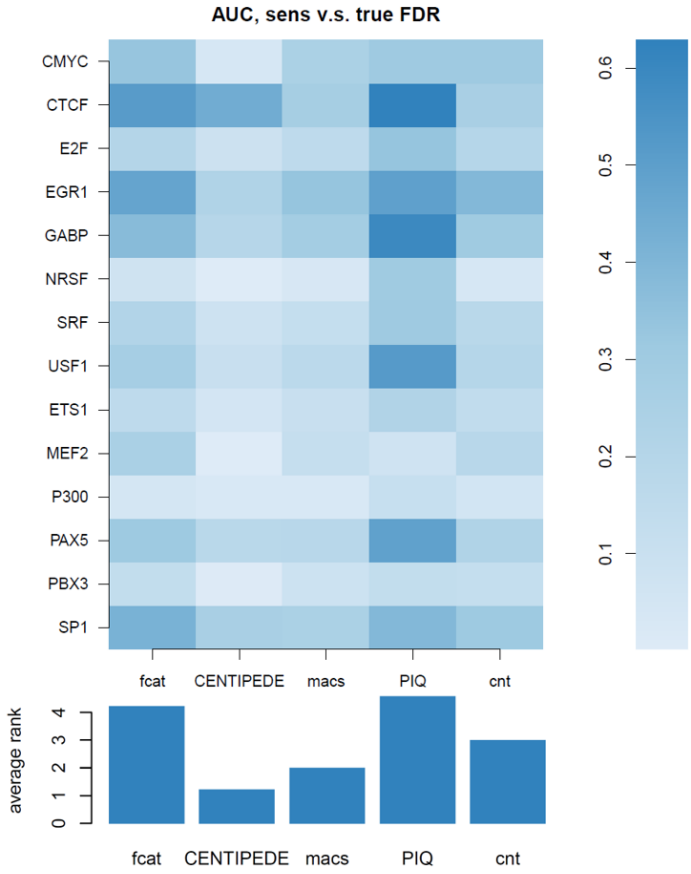

Figure 7.4: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from DNase-Seq in Gm12878

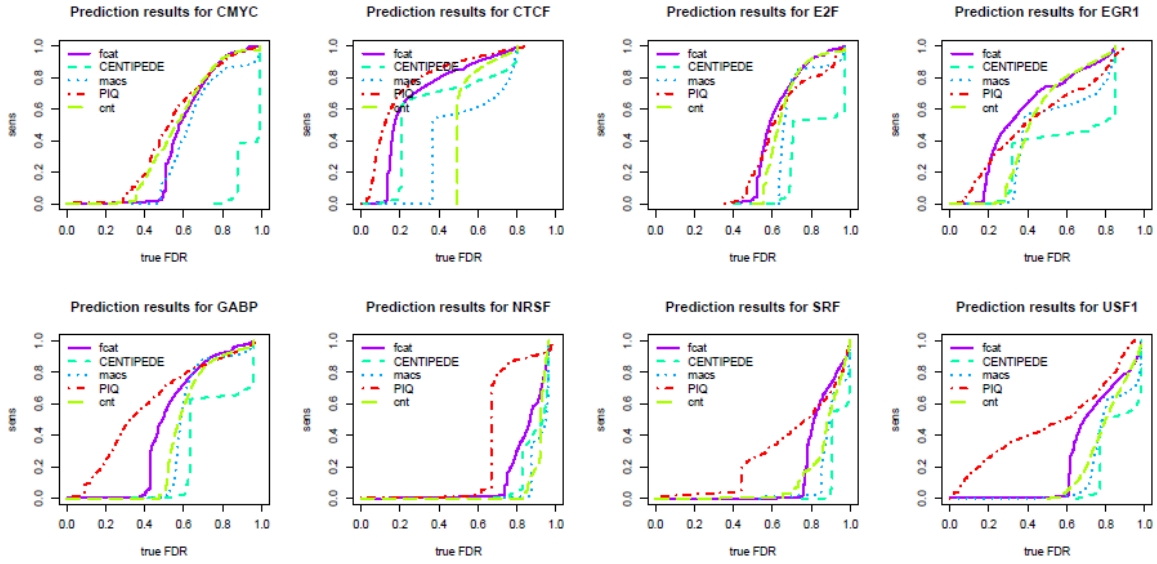

Figure 7.5: Prediction results of sensitivity versus true FDR for TFBS from DNase-Seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in DNase-Seq, CENTIPEDE, MACS, and PIQ in K562.

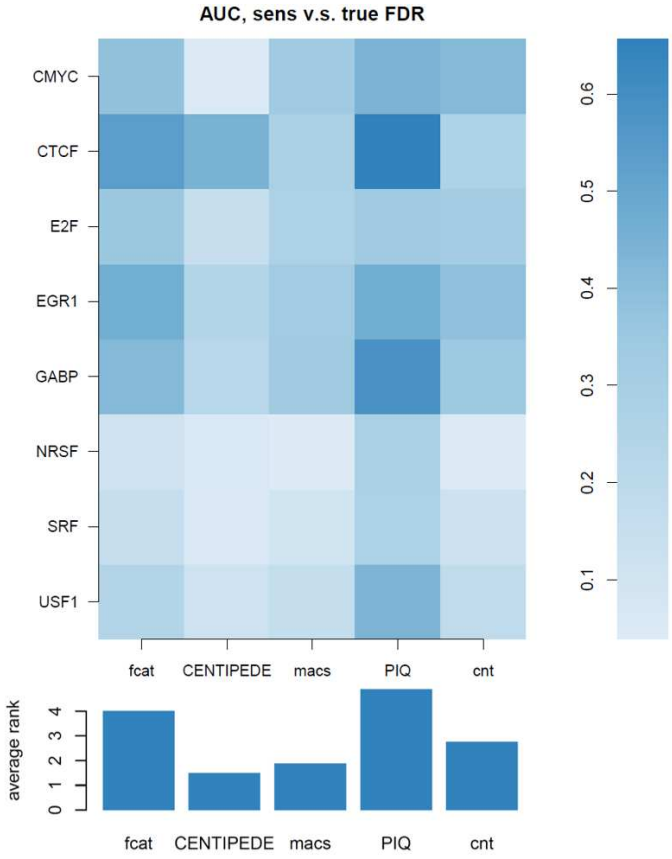

Figure 7.6: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from DNase-Seq in K562

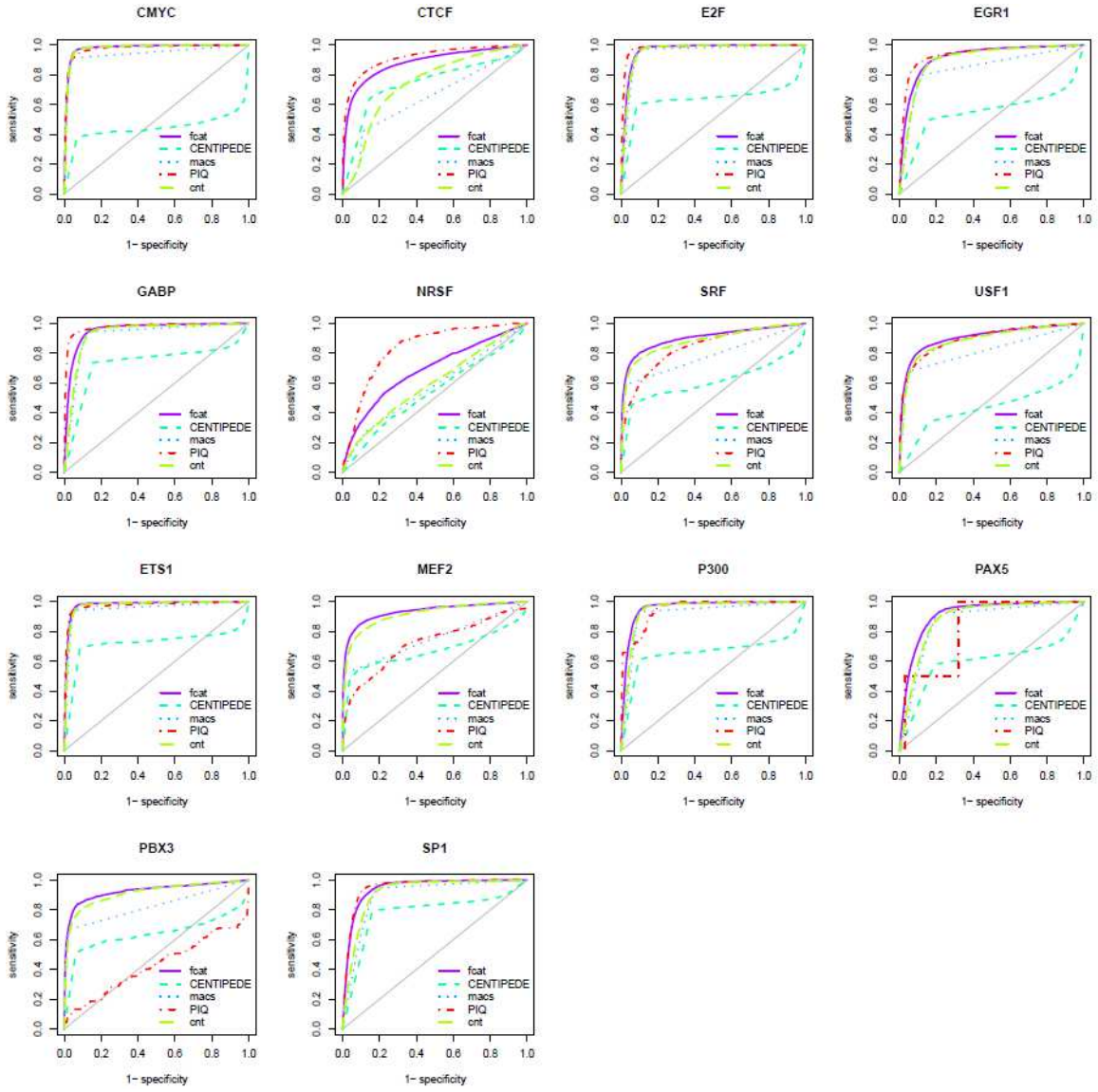

Figure 7.7: TF Prediction results of ROC for TFBS from ATAC-seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in ATAC-seq in Gm12878.

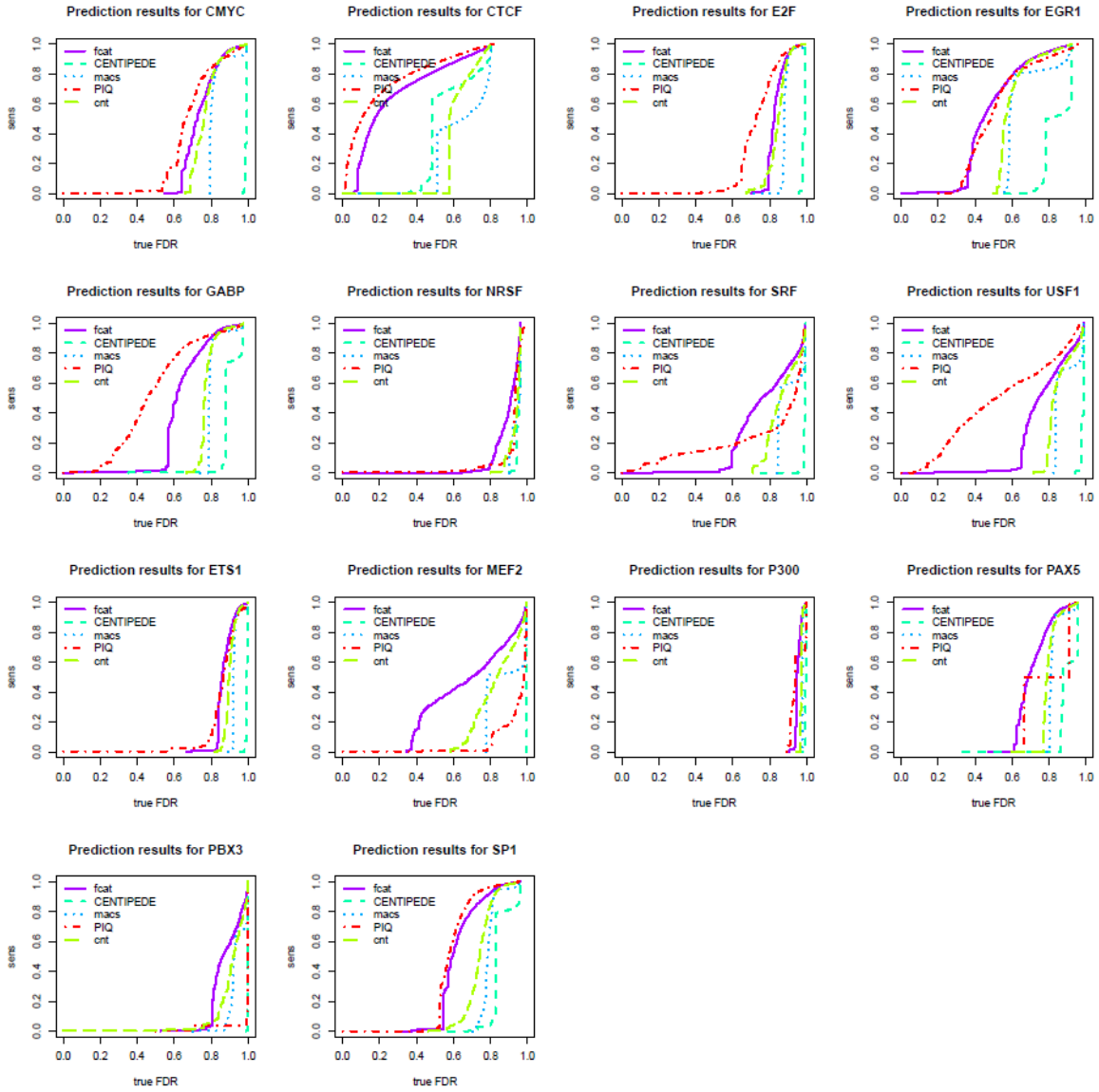

Figure 7.8: TF Prediction results of sensitivity versus true FDR for TFBS from ATAC-seq in Gm12878; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in ATAC-seq in Gm12878.

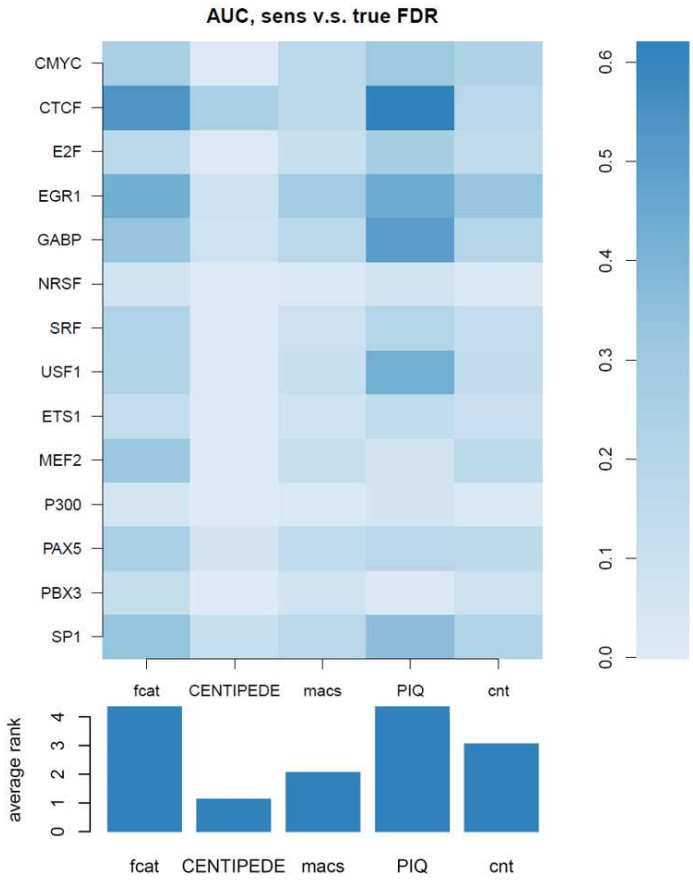

Figure 7.9: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from ATAC-Seq in Gm12878

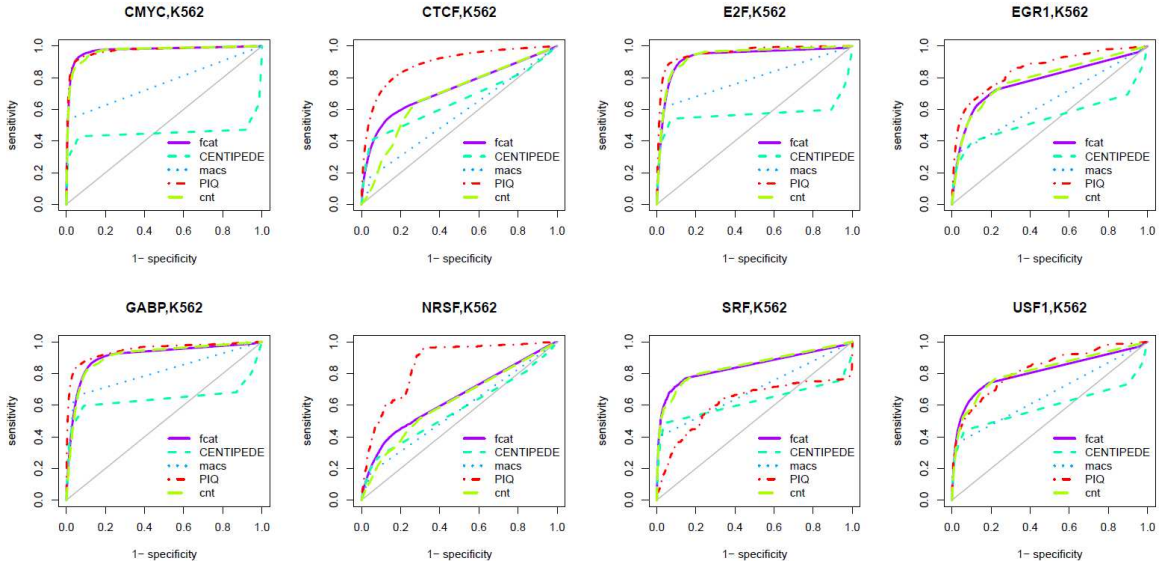

Figure 7.10: TF Prediction results of ROC for TFBS from ATAC-seq; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in ATAC-seq in K562.

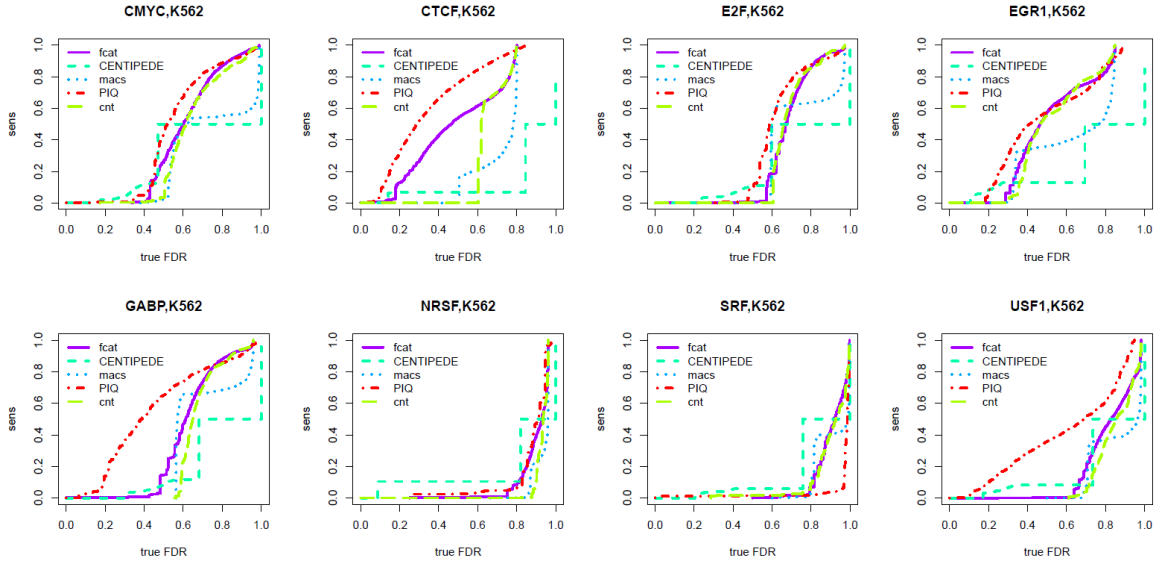

Figure 7.11: TF Prediction results of sensitivity versus true FDR for TFBS from ATAC-seq in K562; each panel shows the sensitivity versus FDR curves for predicting one TF using FCAT in ATAC-seq in K562.

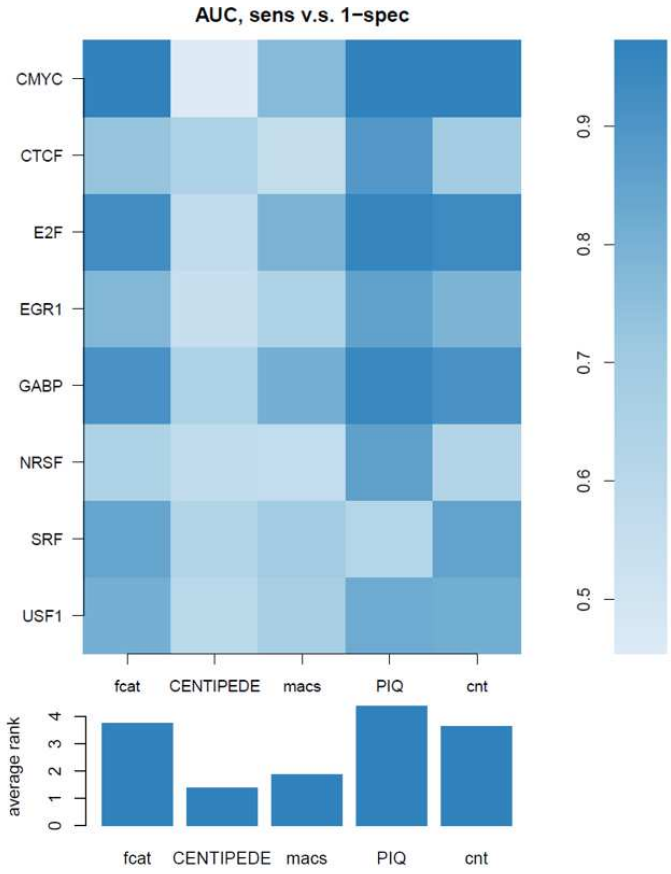

Figure 7.12: AUC and average ranks for prediction results of sensitivity versus 1-specificity for TFBS from ATAC-Seq in K562

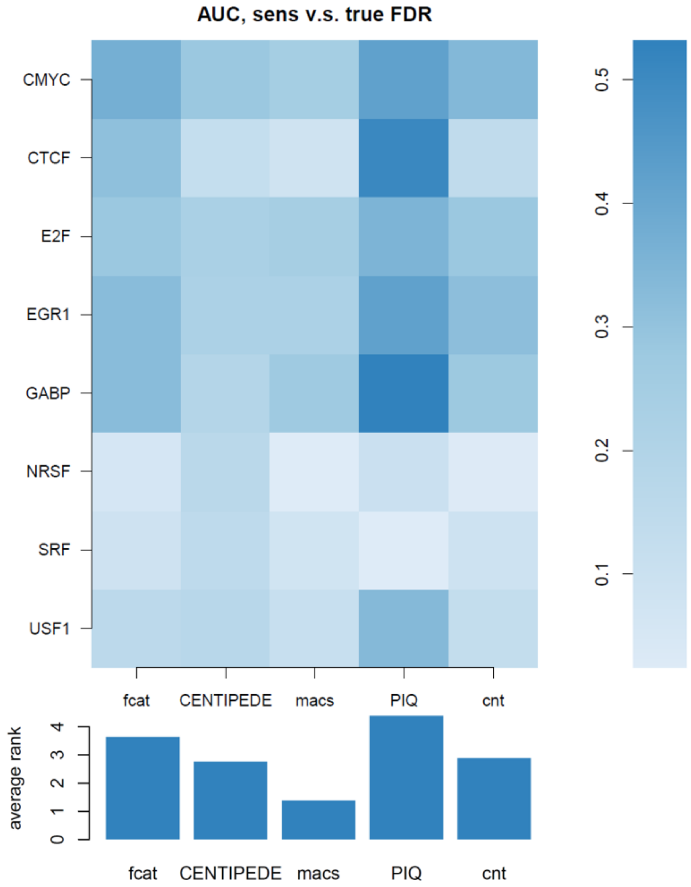

Figure 7.13: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from ATAC-Seq in K562

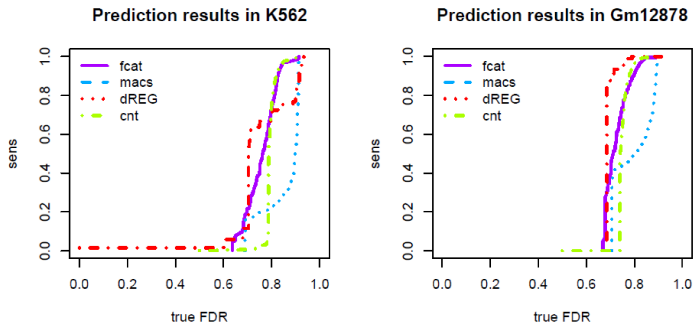

Figure 7.14: Prediction results of sensitivity versus true FDR with GRO-seq in K562 and Gm12878.

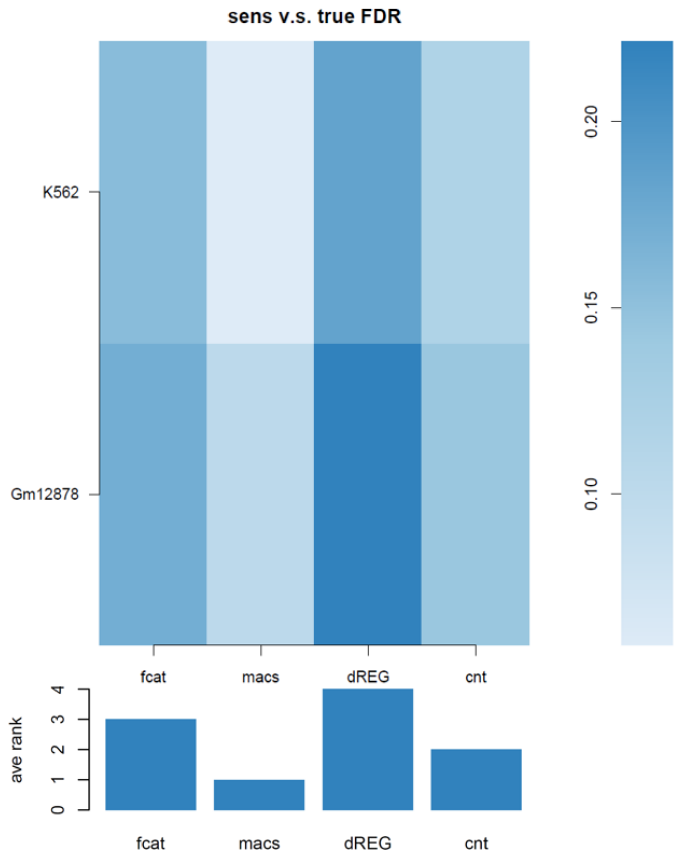

Figure 7.15: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from GRO-seq in K562 and Gm12878

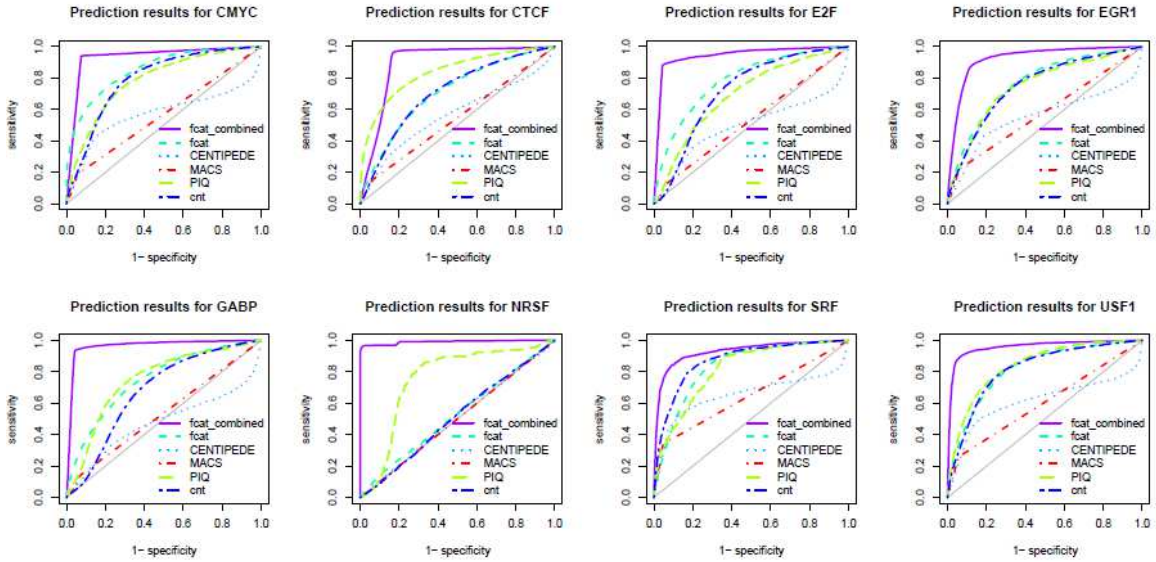

Figure 7.16: TF Prediction results of ROC for TFBS from H3K4me1 combined with historical information

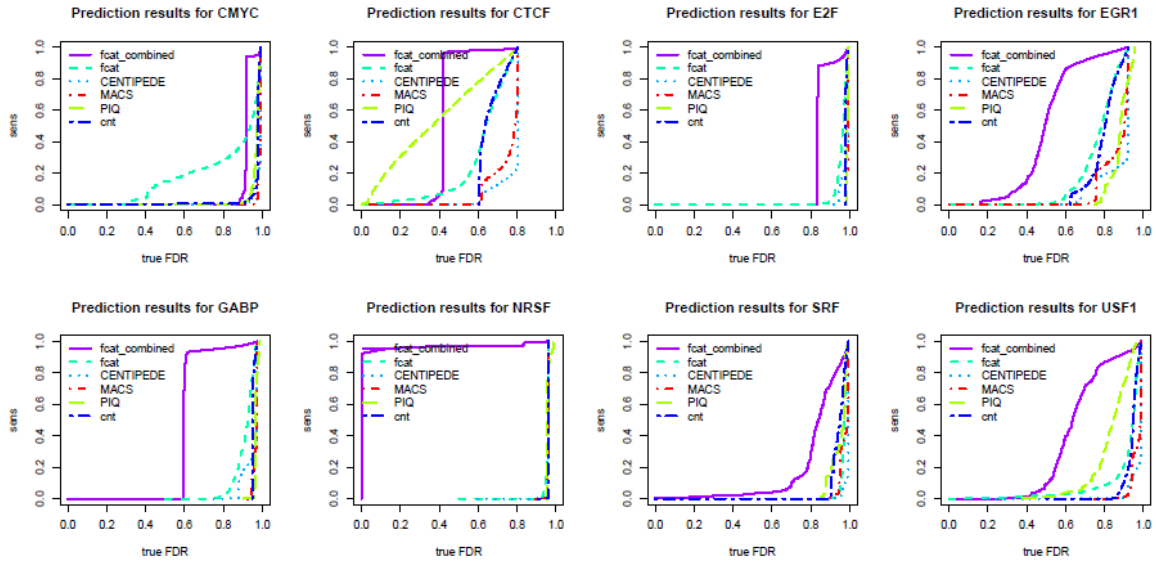

Figure 7.17: TF Prediction results of sensitivity versus true FDR for TFBS from H3K4me1 combined with historical information

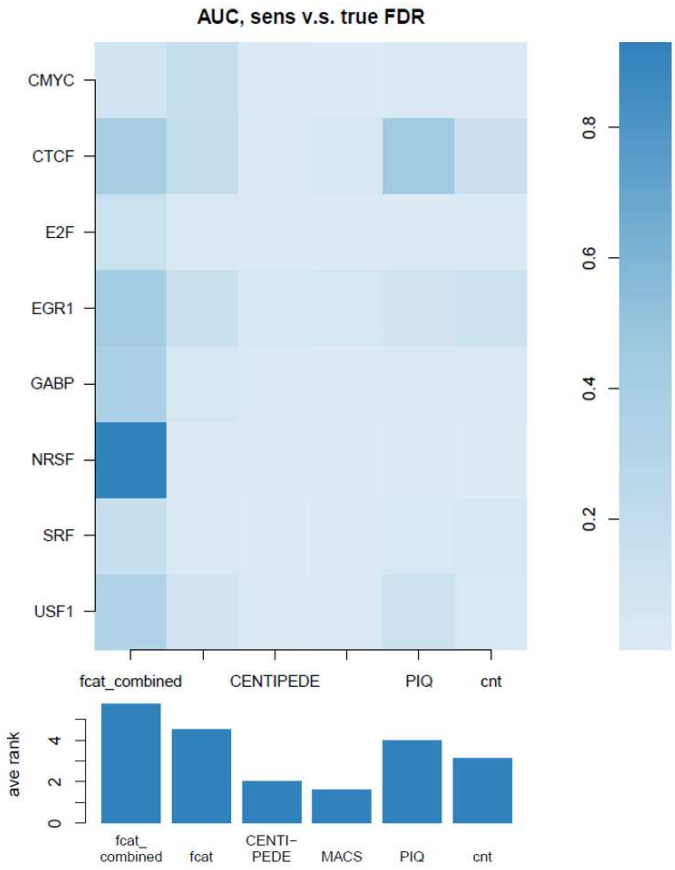

Figure 7.18: AUC and average ranks for prediction results of sensitivity versus true FDR for TFBS from H3K4me1 combined with historical information

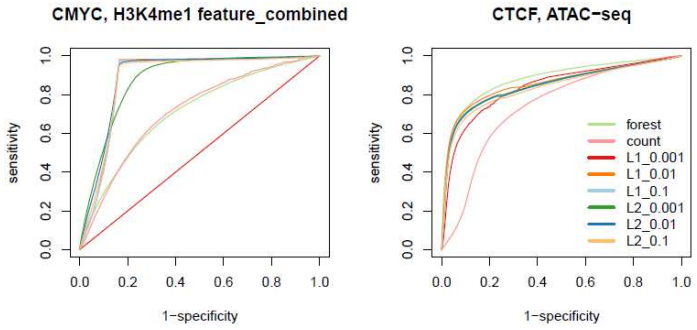

Figure 7.19: Model averaging contributes to FCAT performance in different scenarios. The left panel shows ROC for individual models for using H3K4me1 with combined feature to predict TFBS for CMYC in Gm12878. The right panel gives the ROC of individual models for using ATAC-seq to predict TFBS for CTCF in Gm12878.

Table 7.1: User-specified parameters for extracting features from high-throughput sequencing files in FCAT

Parameters	Values	Description
bin resolution	e.g., 1bp, 5bp, 10bp	The length of bins: log number of reads covering each bin would be used as the bin-wise signal
window size	e.g., 1000bp, 2000bp	The width of window: for a genomic coordinate of interest, all bin-wise signals would be extracted within the window centered at the genomic coordinate as features
paired end	true/false	Indicator for whether the data is paired-end
min fragment length	e.g., 0, 100bp	The minimum length of the fragment that would be counted
max fragment length	e.g., 200bp	The maximum length of the fragment that would be counted

Table 7.2: ENCODE ChIP-seq narrow peak files used for compiling housekeeping motif sites. All files can be download from `http://genome.ucsc.edu/ENCODE/downloads.html`

TFs: ENCODE ChIP-seq narrow peak files
CMYC: wgEncodeAwgTfbsUtaGm12878CmycUniPk.narrowPeak, wgEncodeSydhTfbsA549CmycIggrabPk.narrowPeak, wgEncodeSydhTfbsH1hescCmycIggrabPk.narrowPeak, wgEncodeSydhTfbsHelas3CmycStdPk.narrowPeak, wgEncodeSydhTfbsK562CmycIfna30StdPk.narrowPeak, wgEncodeSydhTfbsK562CmycIfna6hStdPk.narrowPeak, wgEncodeSydhTfbsK562CmycIfng30StdPk.narrowPeak, wgEncodeSydhTfbsK562CmycIfng6hStdPk.narrowPeak, wgEncodeSydhTfbsK562CmycIggrabPk.narrowPeak, wgEncodeSydhTfbsK562CmycStdPk.narrowPeak, wgEncodeSydhTfbsMcf10aesCmycEtoh01HvdPk.narrowPeak, wgEncodeSydhTfbsMcf10aesCmycTam14hHvdPk.narrowPeak, wgEncodeSydhTfbsNb4CmycStdPk.narrowPeak

Table 7.2 ... continued

TFs: ENCODE ChIP-seq narrow peak files
CTCF: wgEncodeSydhTfbsGm12878Ctcfsc15914c20StdPk.narrowPeak, wgEncodeSydhTfbsImr90CtcfbIggrabPk.narrowPeak, wgEncodeSydhTfbsK562CtcfbIggrabPk.narrowPeak, wgEncodeSydhTfbsSknshCtcfbIggrabPk.narrowPeak, wgEncodeUwTfbsA549CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsA549CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAg04449CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAg04449CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAg04450CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAg04450CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAg09309CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAg09309CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAg09319CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAg09319CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAg10803CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAg10803CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsAoafCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsAoafCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsBe2cCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsBe2cCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsBjCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsBjCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsCaco2CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsCaco2CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm06990CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm06990CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12801CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12864CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12864CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12864CtcfStdPkRep3.narrowPeak, wgEncodeUwTfbsGm12865CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12865CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12865CtcfStdPkRep3.narrowPeak, wgEncodeUwTfbsGm12866CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12866CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12867CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12867CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12868CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12868CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12869CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12869CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12870CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12870CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12871CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12871CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12872CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12872CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12872CtcfStdPkRep3.narrowPeak, wgEncodeUwTfbsGm12873CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12873CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12873CtcfStdPkRep3.narrowPeak, wgEncodeUwTfbsGm12874CtcfStdPkRep1.narrowPeak

Table 7.2 ... continued

TFs: ENCODE ChIP-seq narrow peak files
CTCT (cont.): wgEncodeUwTfbsGm12874CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12875CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12875CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsGm12878CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsGm12878CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHacCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHacCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHaspCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHaspCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHbmecCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHbmecCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHcfaaCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHcmCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHcmCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHcpeCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHcpeCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHct116CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHct116CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHeeCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHeeCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHek293CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHek293CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHelas3CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHelas3CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHepg2CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHepg2CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHffCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHffmycCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHffmycCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHl60CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHmecCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHmecCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHmfCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHmfCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHpafCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHpafCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHpfCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHpfCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHreCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHreCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHrpeCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHrpeCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHuvecCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHuvecCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsHvmfCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsHvmfCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsK562CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsK562CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsMcf7CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsMcf7CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsNb4CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsNhdfneoCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsNhdfneoCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsNhekCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsNhekCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsNhlfCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsRptecCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsRptecCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsSaecCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsSaecCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsSknshraCtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsSknshraCtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsWerirb1CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsWerirb1CtcfStdPkRep2.narrowPeak, wgEncodeUwTfbsWi38CtcfStdPkRep1.narrowPeak, wgEncodeUwTfbsWi38CtcfStdPkRep2.narrowPeak

Table 7.2 ... continued

TFs: ENCODE ChIP-seq narrow peak files
E2F: wgEncodeAwgTfbsSydhGm12878E2f4IggmusUniPk.narrowPeak, wgEncodeAwgTfbsSydhHelas3E2f4UniPk.narrowPeak, wgEncodeAwgTfbsSydhHelas3E2f6UniPk.narrowPeak, wgEncodeAwgTfbsSydhK562E2f4UcdUniPk.narrowPeak, wgEncodeAwgTfbsSydhK562E2f6UcdUniPk.narrowPeak, wgEncodeAwgTfbsSydhMcf10aesE2f4TamHvdUniPk.narrowPeak
EGR1: wgEncodeAwgTfbsHaibH1hescEgr1V0416102UniPk.narrowPeak, wgEncodeAwgTfbsHaibK562Egr1V0416101UniPk.narrowPeak
GABP: wgEncodeAwgTfbsHaibGm12878GabpPcr2xUniPk.narrowPeak, wgEncodeAwgTfbsHaibH1hescGabpPcr1xUniPk.narrowPeak, wgEncodeAwgTfbsHaibHelas3GabpPcr1xUniPk.narrowPeak, wgEncodeAwgTfbsHaibHepg2GabpPcr2xUniPk.narrowPeak, wgEncodeAwgTfbsHaibK562GabpV0416101UniPk.narrowPeak

Table 7.3: Position Weight Matrix (PWM) for TFs used in the applications.

A	C	G	T	A	C	G	T	A	C	G	T
CMYC				GABP				USF1			
0.1	21.1	0.1	0.1	4.1	2.6	2.8	0.1	1.1	3.1	7.1	1.1
20.1	0.1	0.1	1.1	0.1	7.3	2	0.1	2.1	4.1	0.1	6.1
0.1	21.1	0.1	0.1	1.3	6.8	1.4	0.1	0.1	12.1	0.1	0.1
0.1	2.1	18.1	1.1	0.1	0.1	9.3	0.1	12.1	0.1	0.1	0.1
0.1	0.1	1.1	20.1	0.1	0.1	9.3	0.1	0.1	12.1	0.1	0.1
1.1	0.1	19.1	1.1	9.3	0.1	0.1	0.1	2.1	0.1	8.1	2.1
0.1	8.1	8.1	5.1	9.3	0.1	0.1	0.1	1.1	1.1	3.1	7.1
	CTCF			0.1	0.1	9.3	0.1	0.1	1.1	10.1	1.1
87.1	291.1	76.1	459.1	1.6	2.3	1	4.6	3.1	1.1	5.1	3.1
167.1	145.1	414.1	187.1	0.1	0.1	8.4	1	0.1	7.1	2.1	3.1
281.1	49.1	449.1	134.1	0.1	6.1	2.2	1.2	ETS1			
56.1	800.1	21.1	36.1	4.5	0.9	3.1	1.1	4.4	2.5	4.1	0.1
8.1	903.1	0.1	2.1	NRSF				0.7	8.3	3.9	0.1
744.1	13.1	65.1	91.1	0.1	2.1	11.1	1.1	14.8	0.1	0.1	0.1
40.1	528.1	334.1	11.1	1.1	8.1	0.1	5.1	0.1	0.1	14.8	0.1
107.1	433.1	48.1	324.1	4.1	1.1	7.1	2.1	0.1	0.1	14.8	0.1
851.1	11.1	32.1	18.1	0.1	13.1	1.1	0.1	14.8	0.1	0.1	0.1
5.1	0.1	903.1	3.1	1.1	0.1	0.1	13.1	10.7	0.1	0.1	4.2
333.1	3.1	566.1	9.1	0.1	2.1	12.1	0.1	3.4	1	10.6	0.1
54.1	12.1	504.1	341.1	0.1	2.1	0.1	12.1	0.1	2.4	2.1	10.5
12.1	0.1	890.1	8.1	0.1	13.1	0.1	1.1	3.5	1.8	8.1	1.8

Table 7.3 ... continued

A	C	G	T	A	C	G	T	A	C	G	T
56.1	8.1	775.1	71.1	0.1	13.1	1.1	0.1	4.8	4	4.1	2.1
104.1	733.1	5.1	67.1	5.1	2.1	7.1	0.1	0.8	3.7	4.7	5.9
372.1	13.1	507.1	17.1	0.1	6.1	0.1	8.1	2.4	0.7	3.1	8.9
82.1	482.1	307.1	37.1	0.1	0.1	12.1	2.1	2.7	4	6.3	2.1
117.1	322.1	73.1	396.1	0.1	2.1	12.1	0.1	3.2	6	5.2	0.7
402.1	181.1	266.1	59.1	1.1	0.1	0.1	13.1	MEF2			
	E2F			0.1	0.1	12.1	2.1	A	C	G	T
1.1	3.1	1.1	17.1	0.1	14.1	0.1	0.1	4.1	0.1	6.1	3.1
1.1	1.1	0.1	20.1	1.1	0.1	0.1	13.1	0.1	2.1	10.1	1.1
1.1	0.1	0.1	21.1	0.1	1.1	13.1	0.1	0.1	9.1	0.1	4.1
0.1	12.1	10.1	0.1	9.1	2.1	3.1	0.1	0.1	1.1	0.1	12.1
0.1	4.1	17.1	0.1	SRF				10.1	1.1	0.1	2.1
0.1	19.1	3.1	0.1	2.1	4.1	11.1	4.1	0.1	0.1	0.1	13.1
0.1	2.1	20.1	0.1	3.1	4.1	7.1	7.1	5.1	0.1	0.1	8.1
0.1	13.1	8.1	1.1	0.1	21.1	0.1	0.1	1.1	0.1	0.1	12.1
	EGR1			0.1	21.1	0.1	0.1	2.1	0.1	0.1	11.1
45.6	6.2	6.2	42.5	19.1	0.1	0.1	2.1	4.1	0.1	0.1	9.1
5.2	2.7	5.2	87.3	9.1	1.1	0.1	11.1	11.1	0.1	2.1	0.1
7.4	0.1	92.8	0.1	19.1	0.1	1.1	1.1	5.1	1.1	5.1	2.1
0.1	98.3	0.1	1.9	1.1	2.1	1.1	17.1	P300			
1.9	0.1	98.3	0.1	17.1	0.1	1.1	3.1	3.1	1.1	3.1	3.1
0.1	1.9	14.6	83.7	10.1	2.1	2.1	7.1	3.1	5.1	2.1	2.1
31	0.1	69.2	0.1	0.1	0.1	21.1	0.1	5.1	3.1	4.1	1.1
0.1	0.1	100.1	0.1	0.1	0.1	20.1	1.1	4.1	1.1	8.1	2.1
0.1	0.1	100.1	0.1	9.1	7.1	5.1	0.1	2.1	0.1	14.1	0.1
12.8	76.5	0.1	11	4.1	9.1	5.1	3.1	2.1	1.1	10.1	3.1
0.1	0.1	100.1	0.1	PAX5				13.1	1.1	1.1	1.1
0.1	0.1	42.4	57.8	2.1	1.1	2.1	0.1	0.1	1.1	14.1	1.1
PBX3				3.1	0.1	2.1	0.1	1.1	0.1	0.1	15.1
0.1	0.1	9.1	0.1	2.1	2.1	0.1	1.1	3.1	1.1	8.1	4.1
8.1	0.1	0.1	1.1	1.1	2.1	2.1	0.1	5.1	4.1	3.1	4.1
0.1	1.1	1.1	7.1	2.1	0.1	1.1	2.1	3.1	3.1	7.1	3.1
1.1	0.1	2.1	6.1	1.1	0.1	4.1	0.1	4.1	4.1	2.1	5.1
1.1	0.1	8.1	0.1	3.1	0.1	1.1	1.1	1.1	5.1	6.1	2.1
9.1	0.1	0.1	0.1	1.1	1.1	1.1	2.1	SP1			
2.1	0.1	1.1	6.1	2.1	0.1	1.1	2.1	32.1	21.1	35.1	20.1
0.1	1.1	5.1	3.1	0.1	2.1	1.1	2.1	24.1	20.1	56.1	8.1
1.1	0.1	6.1	2.1	0.1	3.1	1.1	1.1	14.1	10.1	65.1	19.1

Table 7.3 ... continued

A	C	G	T	A	C	G	T	A	C	G	T
1.1	2.1	2.1	4.1	0.1	1.1	0.1	4.1	17.1	1.1	89.1	1.1
3.1	2.1	1.1	3.1	1.1	1.1	2.1	1.1	0.1	0.1	108.1	0.1
1.1	3.1	4.1	1.1	3.1	0.1	2.1	0.1	0.1	2.1	106.1	0.1
				4.1	0.1	1.1	0.1	19.1	80.1	0.1	9.1
				0.1	0.1	5.1	0.1	2.1	5.1	99.1	2.1
				0.1	5.1	0.1	0.1	0.1	1.1	99.1	8.1
				0.1	0.1	5.1	0.1	21.1	5.1	76.1	6.1
				0.1	0.1	2.1	3.1	17.1	10.1	72.1	9.1
				2.1	0.1	3.1	0.1	3.1	55.1	21.1	29.1
				4.1	0.1	0.1	1.1	9.1	40.1	32.1	27.1
				0.1	5.1	0.1	0.1				
				1.1	2.1	2.1	0.1				
				3.1	0.1	2.1	0.1				
				0.1	2.1	0.1	3.1				
				2.1	1.1	1.1	1.1				
				1.1	2.1	2.1	0.1				
				2.1	2.1	0.1	1.1				

Table 7.4: List of ENCODE files used for gold standards for TFBS

TF	Gm12878 GoldStandard
CMYC	wgEncodeAwgTfbsUtaGm12878CmycUniPk.narrowPeak
CTCF	wgEncodeUwTfbsGm12878CtcfStdPkRep1.narrowPeak,
	wgEncodeUwTfbsGm12878CtcfStdPkRep2.narrowPeak
E2F	wgEncodeAwgTfbsSydhGm12878E2f4IggmusUniPk.narrowPeak
EGR1	wgEncodeAwgTfbsHaibGm12878Egr1Pcr2xUniPk.narrowPeak
GABP	wgEncodeAwgTfbsHaibGm12878GabpPcr2xUniPk.narrowPeak
NRSF	wgEncodeAwgTfbsHaibGm12878NrsfPcr1xUniPk.narrowPeak
SRF	wgEncodeAwgTfbsHaibGm12878SrfPcr2xUniPk.narrowPeak
USF1	wgEncodeAwgTfbsHaibGm12878Usf1Pcr2xUniPk.narrowPeak
EST1	wgEncodeAwgTfbsHaibGm12878Ets1Pcr1xUniPk.narrowPeak
MEF2	wgEncodeAwgTfbsHaibGm12878Mef2aPcr1xUniPk.narrowPeak
P300	wgEncodeAwgTfbsHaibGm12878P300Pcr1xUniPk.narrowPeak
PAX5	wgEncodeAwgTfbsHaibGm12878Pax5c20Pcr1xUniPk.narrowPeak
PBX3	wgEncodeAwgTfbsHaibGm12878Pbx3Pcr1xUniPk.narrowPeak
SP1	wgEncodeAwgTfbsHaibGm12878Sp1Pcr1xUniPk.narrowPeak
TF	K562 GoldStandard
CMYC	wgEncodeSydhTfbsK562CmycStdPk.narrowPeak
CTCF	wgEncodeSydhTfbsK562CtcfbIggrabPk.narrowPeak
E2F	wgEncodeAwgTfbsSydhK562E2f4UcdUniPk.narrowPeak
EGR1	wgEncodeAwgTfbsHaibK562Egr1V0416101UniPk.narrowPeak
GABP	wgEncodeAwgTfbsHaibK562GabpV0416101UniPk.narrowPeak
NRSF	wgEncodeAwgTfbsHaibK562NrsfV0416102UniPk.narrowPeak
SRF	wgEncodeAwgTfbsHaibK562SrfV0416101UniPk.narrowPeak

Table 7.5: List of bam files where features were extracted for TFBS (DNase-seq, Histone ChIP-seq, ATAC-seq)

Data	Cell	Files
DNase-seq	Gm12878	wgEncodeUwDnaseGm12878AlnRep1.bam wgEncodeUwDnaseGm12878AlnRep2.bam;
	K562	wgEncodeUwDnaseK562AlnRep1.bam wgEncodeUwDnaseK562AlnRep2.bam
ChIP-seq	Gm12878	wgEncodeBroadHistoneGm12878H3k4me1StdAlnRep1.bam, wgEncodeBroadHistoneGm12878H3k4me1StdAlnRep2.bam
ATAC-seq	Gm12878	GSE47753; GSM1155957, GSM1155958, GSM1155959, GSM1155960
	K562	GSE65360: Single-cell K562 ATAC-seq from GSE65360 were pooled together as bulk ATAC-seq data for K562

Table 7.6: ENCODE ChIP-seq narrow peak files used for compiling historical information for the application with histone ChIP-seq

TF	ChIP-seq narrow peaks
CMYC	wgEncodeSydhTfbsA549CmycIggrabPk.narrowPeak
	wgEncodeSydhTfbsH1hescCmycIggrabPk.narrowPeak
	wgEncodeSydhTfbsHelas3CmycStdPk.narrowPeak
	wgEncodeSydhTfbsK562CmycStdPk.narrowPeak
	wgEncodeSydhTfbsMcf10aesCmycTam14hHvdPk.narrowPeak
	wgEncodeSydhTfbsNb4CmycStdPk.narrowPeak
CTCF	wgEncodeUwTfbsA549CtcfStdPkRep1.narrowPeak
	wgEncodeSydhTfbsK562CtcfbIggrabPk.narrowPeak
	wgEncodeUwTfbsAg09319CtcfStdPkRep1.narrowPeak
	wgEncodeUwTfbsHelas3CtcfStdPkRep1.narrowPeak
	wgEncodeUwTfbsHuvecCtcfStdPkRep1.narrowPeak
	wgEncodeUwTfbsMcf7CtcfStdPkRep1.narrowPeak
E2F	wgEncodeAwgTfbsSydhMcf10aesE2f4TamHvdUniPk.narrowPeak
	wgEncodeAwgTfbsSydhK562E2f6UcdUniPk.narrowPeak
	wgEncodeAwgTfbsSydhHelas3E2f6UniPk.narrowPeak
EGR1	wgEncodeAwgTfbsHaibK562Egr1V0416101UniPk.narrowPeak
	wgEncodeAwgTfbsHaibH1hescEgr1V0416102UniPk.narrowPeak
GABP	wgEncodeAwgTfbsHaibA549GabpV0422111Etoh02UniPk.narrowPeak
	wgEncodeAwgTfbsHaibHepg2GabpPcr2xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibHelas3GabpPcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibH1hescGabpPcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibK562GabpV0416101UniPk.narrowPeak
NRSF	wgEncodeAwgTfbsHaibA549NrsfV0422111Etoh02UniPk.narrowPeak
	wgEncodeAwgTfbsHaibHelas3NrsfPcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibH1hescNrsfV0416102UniPk.narrowPeak
	wgEncodeAwgTfbsHaibHepg2NrsfV0416101UniPk.narrowPeak
	wgEncodeAwgTfbsHaibK562NrsfV0416102UniPk.narrowPeak
USF1	wgEncodeAwgTfbsHaibA549Usf1Pcr1xDex100nmUniPk.narrowPeak
	wgEncodeAwgTfbsHaibH1hescUsf1Pcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibHepg2Usf1Pcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibK562Usf1V0416101UniPk.narrowPeak
	wgEncodeAwgTfbsHaibSknshraUsf1sc8983V0416102UniPk.narrowPeak
SRF	wgEncodeAwgTfbsHaibH1hescSrfPcr1xUniPk.narrowPeak
	wgEncodeAwgTfbsHaibHepg2SrfV0416101UniPk.narrowPeak
	wgEncodeAwgTfbsHaibK562SrfV0416101UniPk.narrowPeak

Table 7.7: ENCODE files of DNase I uniform narrow peaks and GRO-seq feature files in the application with GRO-seq

Gold Standard	DNaseI hypersensitive sites
Gm12878	wgEncodeAwgDnaseUwdukeGm12878UniPk.narrowPeak
H1hesc	wgEncodeAwgDnaseUwdukeH1hescUniPk.narrowPeak
K562	wgEncodeAwgDnaseUwdukeK562UniPk.narrowPeak
A549	wgEncodeAwgDnaseUwdukeA549UniPk.narrowPeak
Helas3	wgEncodeAwgDnaseUwdukeHelas3UniPk.narrowPeak
Hepg2	wgEncodeAwgDnaseUwdukeHepg2UniPk.narrowPeak
Hmec	wgEncodeAwgDnaseUwdukeHmecUniPk.narrowPeak
Hsmmtube	wgEncodeAwgDnaseUwdukeHsmmtubeUniPk.narrowPeak
Hsmm	wgEncodeAwgDnaseUwdukeHsmmUniPk.narrowPeak
Huvec	wgEncodeAwgDnaseUwdukeHuvecUniPk.narrowPeak
Lncap	wgEncodeAwgDnaseUwdukeLncapUniPk.narrowPeak
Mcf7	wgEncodeAwgDnaseUwdukeMcf7UniPk.narrowPeak
Nhek	wgEncodeAwgDnaseUwdukeNhekUniPk.narrowPeak
Th1	wgEncodeAwgDnaseUwdukeTh1UniPk.narrowPeak
link	`http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeAwgDnaseUniform/`
	GEO access code
Feature	GSE60456; GSM1480326; GSM1480327
link	`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE60456`

Bibliography

[1] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by rna-seq," *Nature methods*, vol. 5, no. 7, pp. 621–628, 2008.

[2] A. Barski, S. Cuddapah, K. Cui, T.-Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao, "High-resolution profiling of histone methylations in the human genome," *Cell*, vol. 129, no. 4, pp. 823–837, 2007.

[3] G. Robertson, M. Hirst, M. Bainbridge, M. Bilenky, Y. Zhao, T. Zeng, G. Euskirchen, B. Bernier, R. Varhol, A. Delaney *et al.*, "Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing," *Nature methods*, vol. 4, no. 8, pp. 651–657, 2007.

[4] L. Song and G. E. Crawford, "Dnase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells," *Cold Spring Harbor Protocols*, vol. 2010, no. 2, pp. pdb–prot5384, 2010.

[5] J. D. Buenrostro, P. G. Giresi, L. C. Zaba, H. Y. Chang, and W. J. Greenleaf,

"Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position," *Nature methods*, vol. 10, no. 12, pp. 1213–1218, 2013.

[6] S. J. Cokus, S. Feng, X. Zhang, Z. Chen, B. Merriman, C. D. Haudenschild, S. Pradhan, S. F. Nelson, M. Pellegrini, and S. E. Jacobsen, "Shotgun bisulphite sequencing of the arabidopsis genome reveals dna methylation patterning," *Nature*, vol. 452, no. 7184, pp. 215–219, 2008.

[7] L. J. Core, J. J. Waterfall, and J. T. Lis, "Nascent rna sequencing reveals widespread pausing and divergent initiation at human promoters," *Science*, vol. 322, no. 5909, pp. 1845–1848, 2008.

[8] Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li *et al.*, "Model-based analysis of chip-seq (macs)," *Genome biology*, vol. 9, no. 9, p. 1, 2008.

[9] A. Valouev, D. S. Johnson, A. Sundquist, C. Medina, E. Anton, S. Batzoglou, R. M. Myers, and A. Sidow, "Genome-wide analysis of transcription factor binding sites based on chip-seq data," *Nature methods*, vol. 5, no. 9, pp. 829–834, 2008.

[10] H. Ji, H. Jiang, W. Ma, D. S. Johnson, R. M. Myers, and W. H. Wong, "An integrated software system for analyzing chip-chip and chip-seq data," *Nature biotechnology*, vol. 26, no. 11, pp. 1293–1300, 2008.

[11] R. Jothi, S. Cuddapah, A. Barski, K. Cui, and K. Zhao, "Genome-wide identification of in vivo protein–dna binding sites from chip-seq data," *Nucleic acids research*, vol. 36, no. 16, pp. 5221–5231, 2008.

[12] H. Xu, C.-L. Wei, F. Lin, and W.-K. Sung, "An hmm approach to genome-wide identification of differential histone modification sites from chip-seq data," *Bioinformatics*, vol. 24, no. 20, pp. 2344–2349, 2008.

[13] P. V. Kharchenko, M. Y. Tolstorukov, and P. J. Park, "Design and analysis of chip-seq experiments for dna-binding proteins," *Nature biotechnology*, vol. 26, no. 12, pp. 1351–1359, 2008.

[14] A. P. Fejes, G. Robertson, M. Bilenky, R. Varhol, M. Bainbridge, and S. J. Jones, "Findpeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology," *Bioinformatics*, vol. 24, no. 15, pp. 1729–1730, 2008.

[15] J. Rozowsky, G. Euskirchen, R. K. Auerbach, Z. D. Zhang, T. Gibson, R. Bjornson, N. Carriero, M. Snyder, and M. B. Gerstein, "Peakseq enables systematic scoring of chip-seq experiments relative to controls," *Nature biotechnology*, vol. 27, no. 1, pp. 66–75, 2009.

[16] Z. S. Qin, J. Yu, J. Shen, C. A. Maher, M. Hu, S. Kalyana-Sundaram, J. Yu, and A. M. Chinnaiyan, "Hpeak: an hmm-based algorithm for defining read-enriched regions in chip-seq data," *BMC bioinformatics*, vol. 11, no. 1, p. 1, 2010.

BIBLIOGRAPHY

[17] P. Agius, A. Arvey, W. Chang, W. S. Noble, and C. Leslie, "High resolution models of transcription factor-dna affinities improve in vitro and in vivo binding predictions," *PLoS Comput Biol*, vol. 6, no. 9, p. e1000916, 2010.

[18] A. Arvey, P. Agius, W. S. Noble, and C. Leslie, "Sequence and chromatin determinants of cell-type–specific transcription factor binding," *Genome research*, vol. 22, no. 9, pp. 1723–1734, 2012.

[19] Y. Guo, S. Mahony, and D. K. Gifford, "High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints," *PLoS Comput Biol*, vol. 8, no. 8, p. e1002638, 2012.

[20] M. Ghandi, D. Lee, M. Mohammad-Noori, and M. A. Beer, "Enhanced regulatory sequence prediction using gapped k-mer features," *PLoS Comput Biol*, vol. 10, no. 7, p. e1003711, 2014.

[21] R. Pique-Regi, J. F. Degner, A. A. Pai, D. J. Gaffney, Y. Gilad, and J. K. Pritchard, "Accurate inference of transcription factor binding from dna sequence and chromatin accessibility data," *Genome research*, vol. 21, no. 3, pp. 447–455, 2011.

[22] A. P. Boyle, L. Song, B.-K. Lee, D. London, D. Keefe, E. Birney, V. R. Iyer, G. E. Crawford, and T. S. Furey, "High-resolution genome-wide in vivo footprinting of diverse transcription factors in human cells," *Genome research*, vol. 21, no. 3, pp. 456–464, 2011.

[23] S. Neph, J. Vierstra, A. B. Stergachis, A. P. Reynolds, E. Haugen, B. Vernot, R. E. Thurman, S. John, R. Sandstrom, A. K. Johnson *et al.*, "An expansive human regulatory lexicon encoded in transcription factor footprints," *Nature*, vol. 489, no. 7414, pp. 83–90, 2012.

[24] R. I. Sherwood, T. Hashimoto, C. W. O'Donnell, S. Lewis, A. A. Barkal, J. P. van Hoff, V. Karun, T. Jaakkola, and D. K. Gifford, "Discovery of directional and nondirectional pioneer transcription factors by modeling dnase profile magnitude and shape," *Nature biotechnology*, vol. 32, no. 2, pp. 171–178, 2014.

[25] C. G. Danko, S. L. Hyland, L. J. Core, A. L. Martins, C. T. Waters, H. W. Lee, V. G. Cheung, W. L. Kraus, J. T. Lis, and A. Siepel, "Identification of active transcriptional regulatory elements from gro-seq data," *Nature methods*, vol. 12, no. 5, pp. 433–438, 2015.

[26] P. N. Cockerill, "Structure and function of active chromatin and dnase i hypersensitive sites," *FEBS journal*, vol. 278, no. 13, pp. 2182–2210, 2011.

[27] A. P. Boyle, S. Davis, H. P. Shulha, P. Meltzer, E. H. Margulies, Z. Weng, T. S. Furey, and G. E. Crawford, "High-resolution mapping and characterization of open chromatin across the genome," *Cell*, vol. 132, no. 2, pp. 311–322, 2008.

[28] R. E. Thurman, E. Rynes, R. Humbert, J. Vierstra, M. T. Maurano, E. Haugen, N. C. Sheffield, A. B. Stergachis, H. Wang, B. Vernot *et al.*, "The accessible

chromatin landscape of the human genome," *Nature*, vol. 489, no. 7414, pp. 75–82, 2012.

[29] G. E. Crawford, I. E. Holt, J. Whittle, B. D. Webb, D. Tai, S. Davis, E. H. Margulies, Y. Chen, J. A. Bernat, D. Ginsburg *et al.*, "Genome-wide mapping of dnase hypersensitive sites using massively parallel signature sequencing (mpss)," *Genome research*, vol. 16, no. 1, pp. 123–131, 2006.

[30] S. John, P. J. Sabo, R. E. Thurman, M.-H. Sung, S. C. Biddie, T. A. Johnson, G. L. Hager, and J. A. Stamatoyannopoulos, "Chromatin accessibility predetermines glucocorticoid receptor binding patterns," *Nature genetics*, vol. 43, no. 3, pp. 264–268, 2011.

[31] A. P. Boyle, J. Guinney, G. E. Crawford, and T. S. Furey, "F-seq: a feature density estimator for high-throughput sequence tags," *Bioinformatics*, vol. 24, no. 21, pp. 2537–2538, 2008.

[32] N. U. Rashid, P. G. Giresi, J. G. Ibrahim, W. Sun, and J. D. Lieb, "Zinba integrates local covariates with dna-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions," *Genome biology*, vol. 12, no. 7, p. 1, 2011.

[33] P. J. Park, "Chip–seq: advantages and challenges of a maturing technology," *Nature Reviews Genetics*, vol. 10, no. 10, pp. 669–680, 2009.

[34] S. J. Wheelan, L. Z. Scheifele, F. Martínez-Murillo, R. A. Irizarry, and J. D. Boeke, "Transposon insertion site profiling chip (tip-chip)," *Proceedings of the National Academy of Sciences*, vol. 103, no. 47, pp. 17 632–17 637, 2006.

[35] E. P. Consortium *et al.*, "The encode (encyclopedia of dna elements) project," *Science*, vol. 306, no. 5696, pp. 636–640, 2004.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[38] S. Le Cessie and J. C. Van Houwelingen, "Ridge estimators in logistic regression," *Applied statistics*, pp. 191–201, 1992.

[39] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, pp. 382–401, 1999.

[40] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the royal statistical society. Series B (Methodological)*, pp. 289–300, 1995.

[41] M. F. Melgar, F. S. Collins, and P. Sethupathy, "Discovery of active enhancers through bidirectional expression of short transcripts," *Genome biology*, vol. 12, no. 11, p. 1, 2011.

[42] R. Andersson, C. Gebhard, I. Miguel-Escalada, I. Hoof, J. Bornholdt, M. Boyd, Y. Chen, X. Zhao, C. Schmidl, T. Suzuki *et al.*, "An atlas of active enhancers across human cell types and tissues," *Nature*, vol. 507, no. 7493, pp. 455–461, 2014.

[43] M. Lizio, J. Harshbarger, H. Shimoji, J. Severin, T. Kasukawa, S. Sahin, I. Abugessaisa, S. Fukuda, F. Hori, S. Ishikawa-Kato *et al.*, "Gateways to the fantom5 promoter level mammalian expression atlas," *Genome biology*, vol. 16, no. 1, p. 1, 2015.

[44] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[45] J. S. Han and J. D. Boeke, "Line-1 retrotransposons: Modulators of quantity and quality of mammalian gene expression?" *Bioessays*, vol. 27, no. 8, pp. 775–784, 2005.

[46] H. H. Kazazian, "Mobile elements: drivers of genome evolution," *science*, vol. 303, no. 5664, pp. 1626–1632, 2004.

[47] C. R. L. Huang, A. M. Schneider, Y. Lu, T. Niranjan, P. Shen, M. A. Robinson, J. P. Steranka, D. Valle, C. I. Civin, T. Wang *et al.*, "Mobile interspersed repeats are major structural variants in the human genome," *Cell*, vol. 141, no. 7, pp. 1171–1182, 2010.

BIBLIOGRAPHY

[48] C. R. L. Huang, K. H. Burns, and J. D. Boeke, "Active transposition in genomes," *Annual review of genetics*, vol. 46, p. 651, 2012.

[49] A. A. Mir, C. Philippe, and G. Cristofari, "eul1db: the european database of l1hs retrotransposon insertions in humans," *Nucleic acids research*, vol. 43, no. D1, pp. D43–D47, 2015.

[50] A. Mathelier, O. Fornes, D. J. Arenillas, C.-y. Chen, G. Denay, J. Lee, W. Shi, C. Shyr, G. Tan, R. Worsley-Hunt *et al.*, "Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles," *Nucleic acids research*, vol. 44, no. D1, pp. D110–D115, 2016.

[51] J.-M. Belton, R. P. McCord, J. H. Gibcus, N. Naumova, Y. Zhan, and J. Dekker, "Hi–c: a comprehensive technique to capture the conformation of genomes," *Methods*, vol. 58, no. 3, pp. 268–276, 2012.

Vita

Bing He received a Bachelor's degree in Information Management and System from Peking University in China in 2009. She also received a Master's degree in Information Science from Indiana University Bloomington in 2011. She joined the Sc.M program in Biostatistics in 2011 and enrolled in the Ph.D program in Biostatistics in 2013. She received the Margaret Rufsvold Fellowship from Indiana University in 2011 and the Kocherlakota Award from the Department of Biostatistics at Johns Hopkins Bloomberg School of Health in 2012. Her research focuses on predictive modeling in bioinformatics and computational biology.